

Actuate e.Reporting Suite 5



Building an e.Reporting Web Site
Release 5

Information in this document is subject to change without notice. Examples provided are fictitious. No part of this document may be reproduced or transmitted in any form, or by any means, electronic or mechanical, for any purpose, in whole or in part, without the express written permission of Actuate Corporation.

© 1995 - 2000 by Actuate Corporation. All rights reserved. Printed in the United States of America.

Contains information proprietary to:

Actuate Corporation
701 Gateway Boulevard
South San Francisco, CA 94080
<http://www.actuate.com>

The software described in this manual is provided by Actuate Corporation under an Actuate License agreement. The software may be used only in accordance with the terms of the agreement.

Actuate Corporation trademarks and registered trademarks:
Actuate, e.Analysis, e.Report, e.Reporting, Live Report Document, Live Report Extension, ReportBlast, ReportCast, Report Encyclopedia, SmartSearch, Transporter, Virtual Report Distribution, and XML Reports.

Third party trademarks:

ActiveX, Microsoft, MSN, The Microsoft Network, Windows, Windows NT, Windows 95/98 and/or other Microsoft products are either trademarks or registered trademarks of Microsoft. CT-Library is a trademark of Sybase, Inc. Databeacon is a trademark of InterNetivity Inc. Digital Creations and DCLC, are trademarks of Digital Creations, LC. Graphics Server is a trademark of Bits Per Second, Ltd. and Pinnacle Publishing, Inc. HDK is a registered trademark of Virtual Media Technology Pty Ltd. INFORMIX-ESQL/C is a trademark of Informix Software, Inc. InstallShield is a registered trademark of InstallShield Corporation. Java, ONC, Solaris, SPARC, Sun, and Sun Microsystems are trademarks of Sun Microsystems. LEADTOOLS is a registered trademark of LEAD Technologies, Inc. Netscape, Netscape Communications, Netscape Communicator, Netscape Enterprise Server, Netscape FastTrack Server, and Netscape Navigator are either trademarks or registered trademarks of Netscape Communications Corporation. NobleNet and WinRPC are trademarks of NobleNet, Inc. Oracle Call Interface is a trademark of Oracle Corporation. Progress is a registered trademark of Progress Software Corporation. Sheridan Calendar Widgets is a trademark of Sheridan Software Systems, Inc. SmartHeap is a trademark of MicroQuill Software Publishing, Inc. Tools.h++ is a trademark of Rogue Wave Software, Inc. UNIX is a registered trademark of X/Open Company, Ltd. XPrinter is a trademark of Bristol Technology, Inc. XPAT, created by James Clark, is licensed under the Mozilla license agreement. WinWidgets is a trademark of Simple Software, Inc.

All other brand or product names are trademarks or registered trademarks of their respective companies or organizations.

Document No. 001211-2-630301 November 13, 2000

Contents

Introduction	vii
About Actuate e.Reporting Suite 5.	vii
About <i>Building an e.Reporting Web Site</i>	x
About Actuate e.Reporting Server product	xi
Online documentation.	xiii
Using online manuals	xiii
Online help	xiii
Using online help	xiii
Using the Actuate online help system.	xiii
Using report-specific online help	xv
Typographical conventions	xvi
Syntax conventions	xvii
 Chapter 1	
Using ReportCast	1
About Actuate ReportCast	2
Setting up ReportCast	4
Navigating the Report Encyclopedia on the Web	4
Accessing the Report Encyclopedia on the Web.	5
Accessing channels and reports.	7
Accessing, generating, and searching reports on the Web.	8
Generating a report.	8
Setting report viewing options	11
Searching DHTML reports.	12
Examining the search URL	15
Showing/Hiding the e.Analysis Button	16
Navigating through the Report Encyclopedia.	16
Working with ReportCast channels	17
Subscribing to a ReportCast channel	17
Unsubscribing from a ReportCast channel.	17
 Chapter 2	
Administering ReportCast	19
About ReportCast administration	20
About ReportCast installations	20
Starting and stopping ReportCast	21
Starting and stopping ReportCast on Windows server systems.	21

Starting and stopping ReportCast on UNIX systems	22
Starting a UNIX ReportCast service	22
Stopping a UNIX ReportCast server	22
Setting ReportCast default e.Reporting Server	23
Changing the Windows servers ReportCast TCP/IP port	24
Changing the ReportCast server machine	25
Using non-default web server ports	26
Configuring ReportCast port numbers	26
Creating custom configurations	28
Mapping custom configurations to locales	29
Supporting locales with e.Reporting Server and ReportCast.	31
Accessing UNIX ReportCast with CGI scripts	31
Working with reports on Windows web sites	32
Using CGI scripts for security.	33
Administering ReportCast channels	34
About Actuate ReportCast channels	34
URLs in e-mail notification	39
Tuning ReportCast performance	39
Setting user connection timeouts	39
Setting request timeouts	40
Tuning viewing performance	40
Viewing files using the native web server file system	41
Configuring viewing options	42
Handling viewing security issues	43

Chapter 3

Working with ReportCast templates 45

About ReportCast templates	46
About HTML template files	46
About internal template structure.	49
About template placement and scope.	49
About the ReportCast object model.	51
ReportCast objects	51
Working with the ReportCast object model.	52
Understanding objects available to a template	53
About the web server template directory structure	53
Creating and customizing ReportCast template files	55
Template file elements.	55
Modifying HTML template files.	56
Locating ReportCast template files	57
Understanding ReportCast searches for templates.	57
Creating custom requester forms for a specific report	59

Reloading template files60
Naming template files61
Placing custom templates in a Report Encyclopedia63
Creating custom templates67
Customizing the e.Analysis add-on option71
Customizing the DHTML and SmartSearch displays.73

Chapter 4

ReportCast directives 75

About ReportCast directives76
Sending directives76
Managing directives77
Formatting directives.78
Types of ReportCast directives80
ReportCast directives.81
Directives syntax.83
Folder directives84
Viewing DHTML reports86
Search directives91
File directives96
Factory directives98
Using submit.	102
Using status directives	108
Using information directives.	110
About Channel directives	111
Listing ReportCast channels.	112
Administering ReportCast channels.	112
Displaying channel contents	113
Creating channels using channel URLs	113
Subscribing to channels using channel URLs	114
Examples of ReportCast directives.	115

Chapter 5

ReportCast scripting language reference 117

About the ReportCast scripting language.	118
ReportCast scripting language reference	118
List command	119
List files command.	122
List versions command	124
List folders command	125
List items command	125
List requestforms command	126

List requests	126
List paramgroups, list parameters commands	127
List values command	129
List status command	130
List members command	131
List grants command	132
List privileges command	133
List notifications command	134
List schedules command	134
List printerprops command	135
List channels command	135
List reportservers command	136
List pathlinks command	136
List expirationpolicies command	138
List tocitems command	138
List criteria command	138
List resulthits command	139
List resultfields command	139
List viewpageformats	140
List searchformats	141
If command	142
Include command	143
With command	143
Set command	144
ReportCast scripting language variables	145
ReportCast variables	146
Web server variable	147
Web request variables	147
e.Reporting Server variables	148
View session variables	149
Table of contents item variables	150
Search criterion variables	150
Search result set variables	151
Search result hit variables	151
Search result field variables	151
Current folder variables	152
Folder item variables	152
Current request folder	154
Report definition	154
Request item variables	155
Member variables	157
Privilege variables	157
Request status variables	158
Parameter group variable	158

Parameter variables.	158
Schedule variable.	159
Printer variables.	159
ReportCast Channel variables.	160
Error variables.	161
 Chapter 6	
Internal ReportCast security	163
About ReportCast security	164
Setting the ReportCast security variables	164
Providing security for ReportCast administration	166
Logging on and authenticating users	166
Logging on as anonymous.	166
Logging on as an e.Reporting Server user	167
Authenticating users using forms.	167
Security and the Apache web server.	168
Customizing the Login dialog box.	168
Setting the idle user session timeout.	168
Modifying the HTTP unauthorized status code	169
About the ReportCast Security Extension.	169
Overview	169
Developing the ReportCast Security Extension	175
ReportCast Agent Security Extension library functions.	176
AcRCAgentAuthenticateRequestBody	176
AcRCAgentAuthenticateRequestHeader	177
ReportCast Server Security Extension library functions	180
AcWebSecurityStart function	180
AcWebSecurityAuthenticate function	181
AcWebSecurityStop function	182
Development considerations	182
NT-specific information	184
Server-specific information	184
Netscape server	184
Microsoft Internet Information server.	184
Enabling RCASE and RCSE.	185
 Chapter 7	
Security for Internet deployment	187
About network security and Actuate	188
Data protection strategies	188
Protecting corporate data	188

Understanding Actuate component communication.	189
Establishing the e.Reporting Server port	190
Configuring the e.Reporting Server port	191
Configuring cached connection management	192
Web browser to web server communication	192
Connecting to and browsing the Report Encyclopedia	192
Running and viewing reports	193
Running reports	194
Configuring data reply ports	194
Viewing e.reports as DHTML, PDF, or XML	196
Configuring the view process port.	197
Viewing e.reports using the LRX.	198
Connecting to the Report Encyclopedia with the split ReportCast configuration	198
Configuring the split ReportCast configuration	199
Deploying Actuate products in Internet environments	200
Understanding configuration options and firewall setup	201
Dual firewalls with ReportCast Agent in the DMZ	202
NAT considerations for dual firewalls with ReportCast Agent in the DMZ	204
Dual firewalls with a proxy server	205
NAT considerations for dual firewalls with a proxy server	206
NAT considerations for the split ReportCast configuration.	206
Placing the e. Reporting Server in the DMZ	206
Index	207

About Actuate e.Reporting Suite 5

Actuate is the leading provider of information delivery solutions for e.Business. e.Business customers use Actuate® e.Reporting Suite 5 to develop and deploy high resolution structured content to hundreds of thousands of users. Actuate takes web reporting to the next level by providing options for needs as varied as seamless personalized web pages and traditional online and printed reports.

Actuate's customer list includes commercial banks, securities, financial services, insurance, high tech, telecom, .com, internet, global 2000, and federal government. OEMs, system integrators, and others building e.Business sites for information delivery face challenges where Actuate e.Reporting Suite 5 offers the following solutions.

Challenge	Actuate solution
Information delivery using HTML	DHTML provides a fast, no download option
Using plug-ins to view structured content	No need to support installations for hundreds of thousands of users
Compromising information display due to lack of integrated tools	Provides template based design and display
Exploding use of web-based content delivery	Ability to support a million hits per day per CPU
Demand for personalized secure information delivery	Open security directory integration and page security
Reuse of existing integrated content	Open server provides access to content from other applications

Challenge	Actuate solution
Maintaining data integrity on hard copy	PDF provides high resolution printed copy
Portability of information into other applications	XML output provides access to data across applications

Actuate tools and reports:

- Solve complex data access problems
- Solve formatting problems that go beyond the scope of other tools
- Scale to support hundreds of thousands of users

The following summary describes the products in the Actuate e.Reporting Suite 5.

Product name	Use
Actuate e.Report Designer Professional	<p>An object-oriented application used by professional developers of structured content to design, build, and distribute report object designs for delivery on the web. The Actuate Basic Language and Actuate Foundation Class Library support extensive customization capabilities.</p> <p>Actuate ActiveX Controls embed Actuate reporting functionality into custom applications.</p> <p>Actuate Requester API accesses attributes and values of report parameters, changes the values of report parameters, controls how and when a report is generated, displays and prints reports, and configures report print setup. Access the Requester API using Actuate Basic, Visual Basic, C, or C++.</p> <p>Actuate search extension API supports developing search extensions to transfer data to any third-party productivity or analysis tool.</p>

Product name	Use
	<p>Actuate report server API implements common Report Encyclopedia tasks, integrates report server features into existing corporate applications, automates routine or time-consuming tasks, and implements new feature groupings for custom business processes. Access the report server API using C++.</p> <p>Actuate Report Server Security Extension supports the use of third-party security tools.</p> <p>Actuate archive driver supports the use of third-party archiving software and hardware.</p>
Actuate e.Report Designer	An application that complements e.Report Designer Professional and is used by business users to design and distribute a variety of reports. These reports require no programming. This application supports both modification of complex reports and using sophisticated components from libraries.
Actuate e.Report Designer Java™ Edition	An application that complements e.Report Designer Professional and e.Report Designer and is used by Java developers to design and distribute a variety of reports. e.Report Designer Java Edition includes both AWT and SWING APIs.
Actuate e.Analysis	<p>An application used to transform data from an Actuate e.report into interactive information. Users can view and analyze data to determine relationships and trends.</p> <p>Actuate e.Analysis is installed on the e.Reporting Server.</p>
Actuate End User Desktop	An application used by end users to request, generate, view, and print report documents. The ReportQuery™ capabilities enable seamless transfer of data from an Actuate report to any productivity tool or analysis tool.

Product name	Use
Actuate Viewer	Application for end users to find, view, and print report documents. The ReportQuery™ capabilities are also part of the Actuate Viewer.
Actuate Live Report Extension (LRX)	Application for end users that works with both Microsoft Internet Explorer and Netscape Navigator to support report viewing and printing on the Web.
Actuate e.Reporting Server	<p>A server application that generates live report documents, manages them in the Report Encyclopedia®, and makes them available to users.</p> <p>This product includes Actuate Administrator Desktop, an application for system and network administrators to manage and control one or more Actuate report servers.</p> <p>This product also includes Actuate ReportCast™ that transforms the Report Encyclopedia into a dynamic, secure web site. ReportCast provides the foundation for Channels and seamless integration with other web sites.</p>
Actuate Advanced e.Reporting Server	<p>An application that adds two capabilities to the basic e.Reporting Server, open server and page security.</p> <ul style="list-style-type: none"> ■ Open server supports the use of third-party report generators with the Actuate e.Reporting Server. ■ Page security supports personalized viewing of parts of reports for different users

Actuate Viewer and Actuate Live Report Extension (LRX) are included with all products except Actuate e.Report Designer Java Edition.

About *Building an e.Reporting Web Site*

Building an e.Reporting Web Site is a guide to providing access to the Actuate Report Encyclopedia on the World Wide Web.

Building an e.Reporting Web Site includes the following chapters:

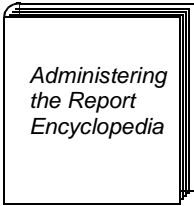

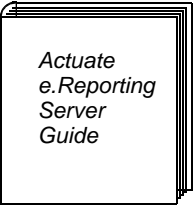

- *Introduction.* This chapter provides an overview of this guide, the Actuate e.Reporting Suite, and the typographical conventions used.
- *Chapter 1. Using ReportCast.* This chapter introduces the e.reporting web site, and explains what you need to deploy, navigate, and access e.reports over the Web. This chapter describes how to work with ReportCast channels.
- *Chapter 2. Administering ReportCast.* This chapter describes how to administer ReportCast. This chapter explains how to start and stop ReportCast, configuration tasks, using CGI with ReportCast, and tuning ReportCast performance.
- *Chapter 3. Working with ReportCast templates.* This chapter describes ReportCast templates, and explains how to create custom templates.
- *Chapter 4. ReportCast directives.* This chapter describes the ReportCast directives.
- *Chapter 5. ReportCast scripting language reference.* This chapter describes ReportCast template scripting language and ReportCast scripting language variables.
- *Chapter 6. Internal ReportCast security.* This chapter describes the ReportCast user authentication and security facilities.
- *Chapter 7. Security for Internet deployment.* This chapter describes security issues for deploying e.reporting web sites over the internet.

About Actuate e.Reporting Server product

Actuate e.Reporting Server documentation includes printed manuals, installation guides, online help, user documentation in PDF and HTML format, and release notes. Information about the product that could not be included before the book printing deadline is in the release notes.

The Actuate web site, <http://www.actuate.com>, contains late-breaking news about the product and its features, as well as product update information. To obtain the password necessary to access the portion of the web site available only to customers, telephone Actuate Customer Support. The engineers in Actuate Customer Support can also help you with technical questions about the product according to your service contract. The Customer Support telephone number, fax number, and e-mail information can be found among the printed materials in the product box.

The printed documentation includes the following manuals.

For information about	See the following
Installation	Installation guides
Late-breaking information about the software and documentation	Release notes
Users, groups, privileges, and roles Printers and printing requests Process groups Building and managing your first Report Encyclopedia	
Viewing, running, finding, and printing reports	
Overview of the Actuate report server architecture Report server administration Database connections Report Encyclopedia utilities	
Actuate ReportCast, its templates, URL directives, scripting language, security and user authentication facilities.	

Online documentation

The information in the printed manuals is also available as online books in Adobe Acrobat PDF format and in the online help system for the Actuate products. For products without a Windows interface such as Actuate e.Reporting Server, Actuate ReportCast, and Actuate e.Analysis, we provide HTML help files. The HTML files install automatically with the product. These files can be viewed with standard browsers.

Using online manuals

The online manuals install automatically with the product. On the product CD, you can also find those files in the Manuals directory. Open the introductory .PDF file to get an overview of the manuals. The items in the table of contents and the page numbers in the index both contain links to the appropriate topics in the text. In the index, you access the link by positioning your cursor over the page number, not the topic.

Online help

Actuate products provide both context-sensitive online help about the product and report-specific online help about the report you are viewing. Actuate e.Reporting Suite makes it possible for developers to create customized report-specific online help.

Using online help

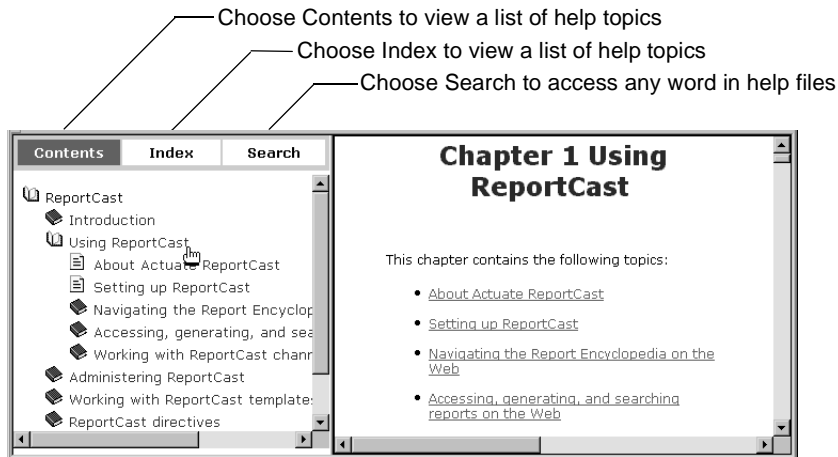
To access online help, choose Help in the toolbar as shown in the following illustration.



Choose a link to access the online help files, or under PDF files, choose a link to the Adobe Acrobat PDF files.

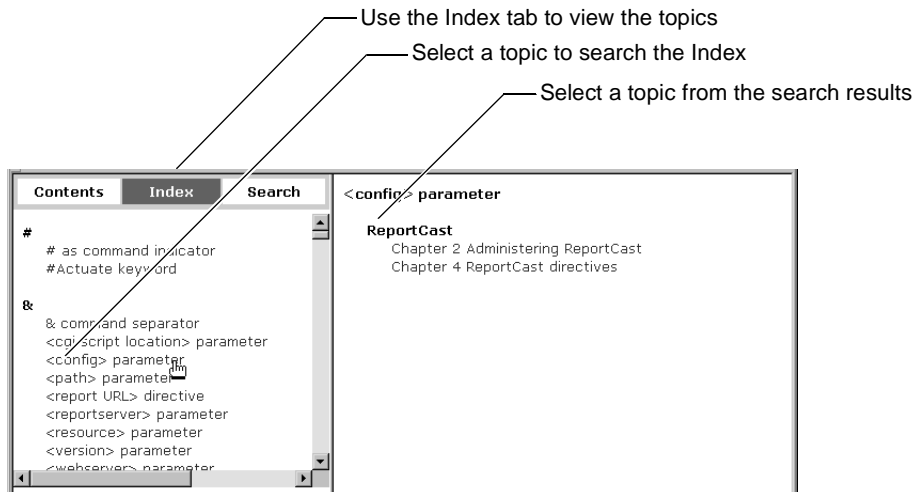
Using the Actuate online help system

Use two windows to access and view information in the e.Reporting Suite help system. The window on the left displays the table of contents or the index of the online help system. The window on the right displays the contents of the online help topics.

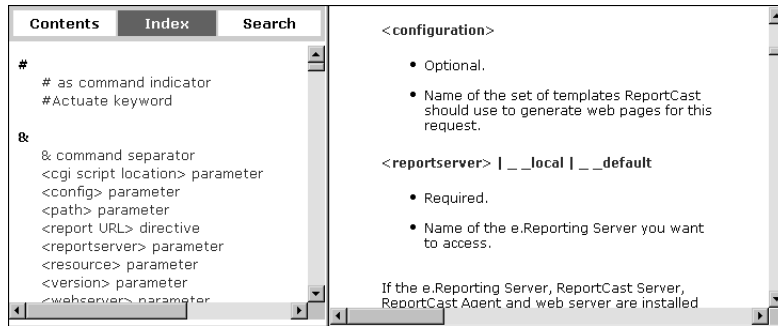


The tabs at the top of the left window access different views. Use these tabs to switch views among the Table of Contents, Index, and Search. The Table of Contents provides an overview of the help file contents. For example, in the preceding window, you see the introduction to the documentation for the e.Reporting Server product. The Search tab accesses any word in the help files.

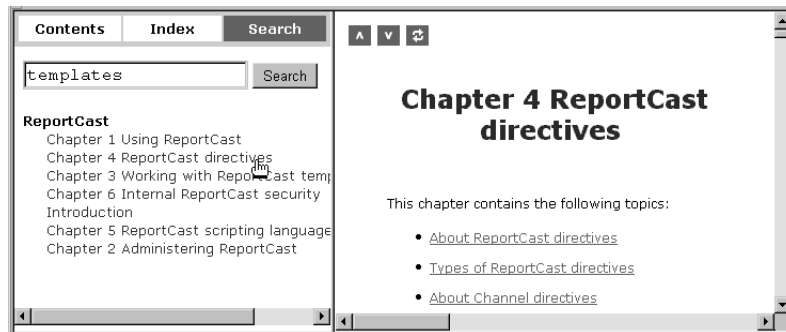
The following two illustrations show an example of the Index and the result of an Index search. The following illustration shows the result of the search as it appears in the window on the right.



To view the topic double-click the topic in the list. The topic displays in the window on the right.

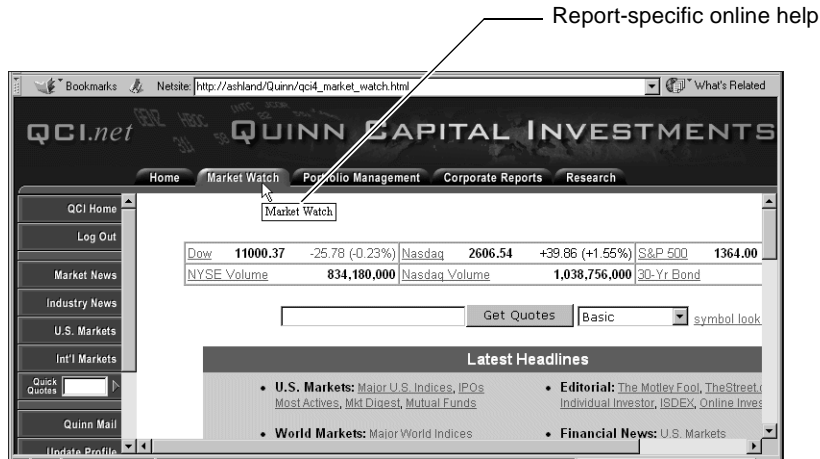


Use the Search tab to find all instances of a keyword in the help files. The following illustration shows the results of a search for the keyword, templates. The topics containing the keyword appear under Search. Double-click a topic in the Search list to display the topic in the right window.



Using report-specific online help

During the design phase, report developers have the option to include report-specific online help. For example, the report developer can add comments to give further detail about specific report objects or to explain how calculations were created.



For detailed information about report-specific online help see Chapter 3, "Viewing a report from the desktop," in *Using e.Reports*.

Typographical conventions

The following table describes the typographical conventions.

Item	Convention	Example
Code examples	Sans serif	Dim As String
File names	All uppercase letters except e. Report Designer Java Edition where file names are case sensitive	DETAIL.ROI
Key combination	A + sign between the keys means to press both keys at the same time	Ctrl+Shift
Menu items	Capitalized. No bold	File
Submenu items	Separated from main menu item with small arrow	File→New
User input or user response	Enclosed in quotation marks	"M*16"

Syntax conventions

The following table describes the symbols used to present the syntax of Actuate Basic language elements.

Symbol	Description	Example
[]	Optional item	[Alias<alias name>]
<>	Argument you must supply	<expression to format>
{ }	Groups two or more mutually exclusive options or arguments	{While Until}
	Separates mutually exclusive options or arguments in a group	Exit {Do For Function Sub}

Using ReportCast

This chapter contains the following topics:

- About Actuate ReportCast
- Setting up ReportCast
- Navigating the Report Encyclopedia on the Web
- Accessing, generating, and searching reports on the Web
- Working with ReportCast channels

About Actuate ReportCast

Actuate ReportCast provides access to the Actuate e.Reporting Server from the World Wide Web. Using ReportCast, you can access and work with reports in the Actuate Report Encyclopedia with any web browser.

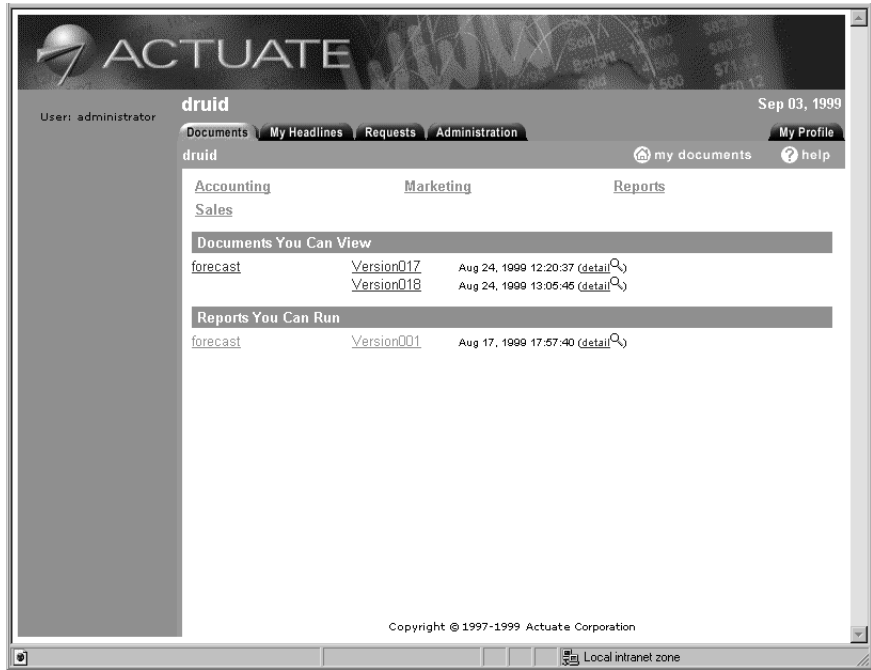
ReportCast technology consists of:

- **ReportCast Agent**—ReportCast's interface to the web server. ReportCast Agent is a dynamic-link library (DLL) that runs as part of your web server. There is no user interface to the DLL.
- **ReportCast Server**—ReportCast's interface to the Actuate e.Reporting Server. ReportCast Server is a service and runs as a process separate from the web server and ReportCast Agent. There is no user interface to the ReportCast Server service.

Previously, to view Actuate reports over the Web you used Actuate LRX (Live Report Extension) for Netscape or Microsoft plug-ins installed with your web browser. With ReportCast, you can view e.reports directly using any web browser. The navigation options of the two viewing methods differ slightly.

ReportCast generates HTML pages that are adaptations of the Actuate Navigator interface, then sends those HTML pages to the web browser. For example, an HTML form that accepts parameter values for report generation replaces the Requester dialog box of the Navigator. Using ReportCast technology you can access HTML reports and DHTML reports without needing any additional software installed on your desktop system.

The following illustration shows the default reporting web site.



The URL format of the default reporting web site is:

`http://<webserver>/acweb/<reportserver>`

- <web server> is the name of the web server running the web interface to the Report Encyclopedia.
- acweb is a keyword indicating that you want to access the Report Encyclopedia.
- <report server> is the name of the e.Reporting Server with the encyclopedia you want to access.

Users navigate the reporting web site by clicking links on the reporting web page, or by entering ReportCast directives as part of a URL.

ReportCast provides a set of default web page templates that determine the appearance of the pages that display in the browser. Commands can be written in a combination of the ReportCast scripting language, Javascript, and standard HTML tags and are embedded in the template. Webmasters have these options for using the templates:

- Use the default templates
- Customize the default templates
- Create entirely new templates

Setting up ReportCast

As you create structured content from databases, you deploy e.reports over the World Wide Web. A typical setup includes:

- A web server
- An Actuate installation consisting of:
 - e.Reporting Server
 - An e.report development tool such as e.Report Designer Professional or e.Report Designer
 - Administrator Desktop
- ReportCast

A webmaster must have administrator privileges to the web server and report server to effectively manage the e.reporting web site.

Navigating the Report Encyclopedia on the Web

Actuate's web interface to the Report Encyclopedia allows you to access and work with reports on the Report Encyclopedia using a web browser. The interface provides web-based Actuate Navigator functionality similar to that provided by the End User Desktop and Administrator Desktop. The web interface also provides a service called ReportCast channels. ReportCast channels use push technology to provide reports to particular web channels. By subscribing to the channels that interest you, you can easily find and view reports on topics of particular importance to you.

Check with your e.Reporting Server administrator to find out if you can access reports on the Web.

To navigate through the Report Encyclopedia using a web browser, click hyperlinks contained on the Encyclopedia's web pages, or enter special URLs or directives, in the Address or Location window of your web browser. For more information about directives, see Chapter 4, "ReportCast directives."

For more information about navigating the Report Encyclopedia, see Chapter 2, "Working with the desktop Navigator," in *Using e.Reports*, the LRX online help, or the online manual in .PDF format. The illustrations in this chapter use the standard web pages provided by Actuate. Your e.Reporting Server administrator may customize the design of the web pages for your site, so your pages may appear different from those in the illustrations.

Accessing the Report Encyclopedia on the Web

To access your Report Encyclopedia, bring up your web browser and enter a URL similar to the following:

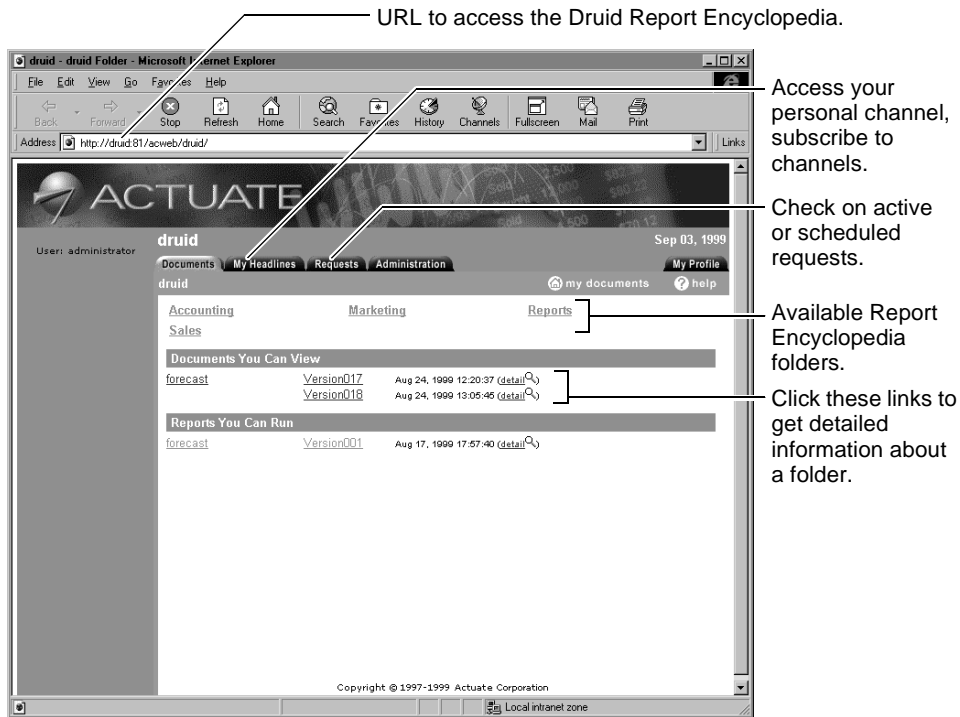
`http://<web server>/acweb/<report server>`

- `<web server>` is the name of the web server running the web interface to the Report Encyclopedia.
- `acweb` is a keyword indicating that you want to access the Report Encyclopedia.
- `<report server>` is the name of the e.Reporting Server with the encyclopedia you want to access.

For example, to access reports on an e.Reporting Server called `Druid` from a web server called `Druid`, running on port 81:

`http://druid:81/acweb/druid`

Depending upon your setup, a Login dialog box may appear. If it does, enter the user name and password you normally use to access your Report Encyclopedia. A Report Encyclopedia web page displays in the browser.



From this main page, you can:

- View the contents of the Report Encyclopedia.
- Access your personal channel (your completed reports folder).
- Access the available Report Encyclopedia folders.
- Get detailed information about folders.
- Sort the folders by name, type, or creation date and time.
- Request report runs.
- Get status information about requests to run reports.

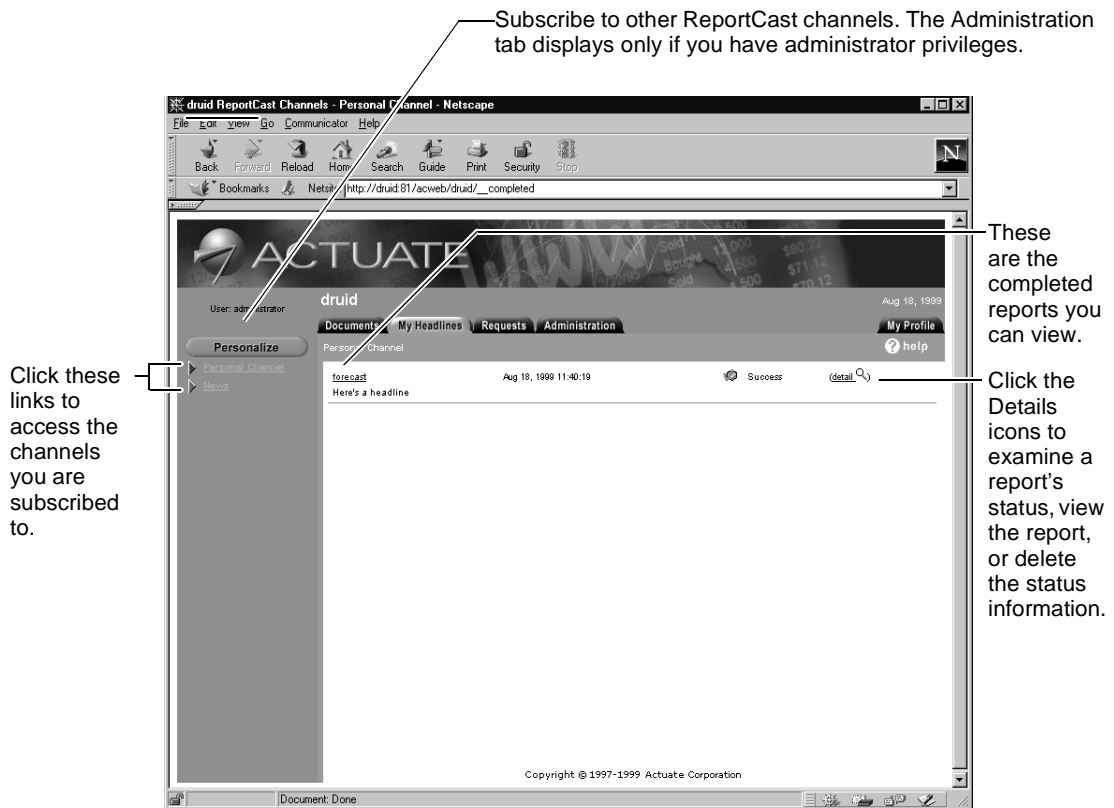
Requests to run reports or to get status information about requests time out after 120 seconds if ReportCast cannot contact the e.Reporting Server.

To ensure that you always see the most current web reporting page, set your web browser to check the date of web pages every time you issue a request to the e.Reporting Server.

To see changes to the Report Encyclopedia that do not involve adding or deleting an item or folder (for example, updating an existing item), force-reload the web reporting page. To force-reload a web reporting page in Netscape, hold the shift-key down at the same time as clicking the reload button. In Microsoft Internet Explorer, clear the cache and then reload the page.

Accessing channels and reports

When you click the My Headlines tab on the main Report Encyclopedia web page, a page such as the following displays in the browser.



From My Headlines page, you can:

- Click the Personalize button to subscribe to and unsubscribe from ReportCast channels.
- Click the channel icons to access particular channels to which you are subscribed.

- View completed reports.
- Click the Details icon to view a report's status, view the report, or delete status information for the report.

Accessing, generating, and searching reports on the Web

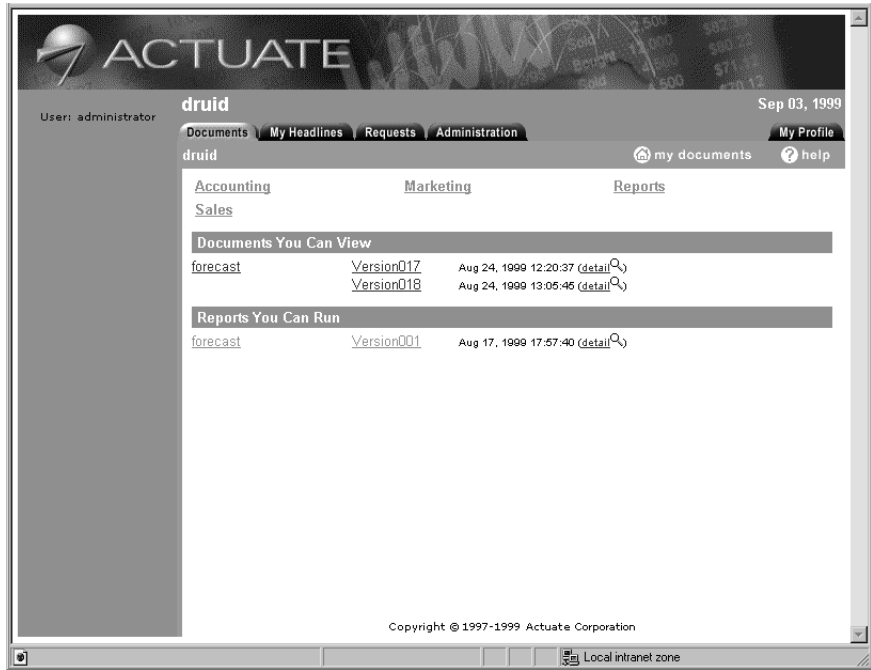
To access and generate reports on the Web:

- Click report executable links in a Report Encyclopedia folder to generate a report.
- Click report links on the My Headlines web page or from other ReportCast channels' pages to view a report.
- Navigate through the Report Encyclopedia folders from the main Report Encyclopedia web page.

Generating a report


When the main web page for the Report Encyclopedia displays, the same list of folders that you normally see in the Administrator Desktop's Navigator window displays. Click a folder's link to display its contents.

Folder contents pages are similar to the following image.



The folder web page lists the folder's contents, including any subfolders, as well as links to your personal folder and to the Active and Scheduled requests web pages.

The difference between this page and your personal channel web page is that the folder contents page lists report executables. To generate a report, click the report executable's link. A requester page similar to the following displays in the browser.


ACTUATE

User: administrator
 Aug 30, 1999

[Documents](#)
[My Headlines](#)
[Requests](#)
[Administration](#)
[My Profile](#)
[? help](#)

[Submit Request](#)

druid
New Request

Parameter Groups

Output Parameters
 Headline (String)

Request
 Wait for results ☒ (when running now)

Advanced

Schedule
☒ Right Now
☐ Once (mm/dd/yyyy) at (hh:mm:ss)
☐ Recurring at (hh:mm:ss)

Priority
☐ High (800) ☒ Medium (500) ☐ Low (200) ☐ Other (1-1000)

Version
☒ Create new version ☐ Keep only the latest version(s)
☐ Overwrite existing version
 Version Name:
 Output Name:

Archive Policy
☒ Use the archive policy for the distribution folder(s)
☐ Delete objects older than days hours
☐ Delete objects on (mm/dd/yyyy) at (hh:mm:ss)
☐ Archive items before deletion (only applies if not using archive policy for distribution folder)

Notify Channels
☐ News

[Submit Request](#)

Copyright © 1997-1999 Actuate Corporation

Fill out the requester fields as you would in the Administrator Desktop, then click the Submit Request button to submit the request. A confirmation form displays in the browser, similar to the following.



Click the Request Status button to check on your request's status. To cancel the request, click the Cancel Request button.

Setting report viewing options

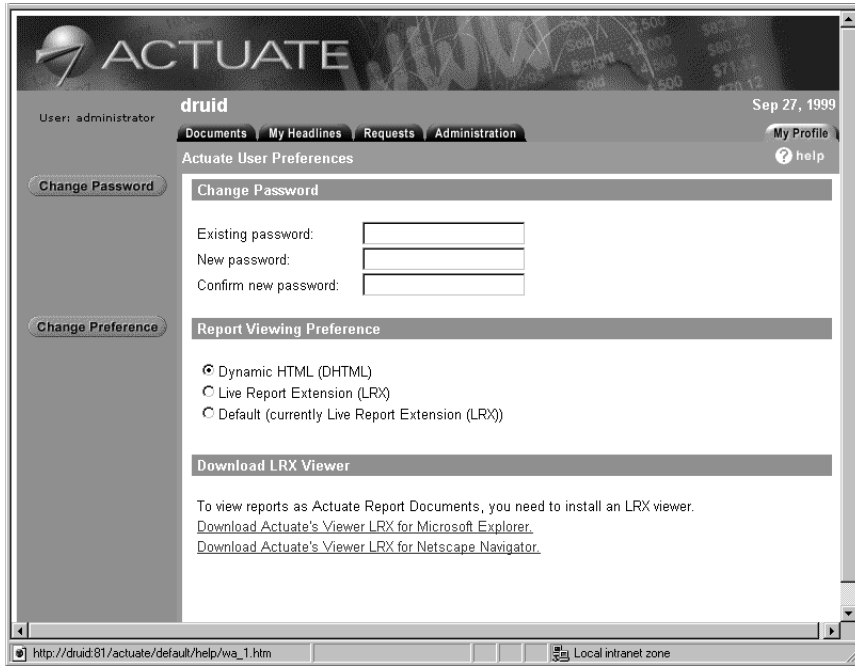
You can tell ReportCast how you want to view reports. Use the User Preferences page to set your report viewing options. Either click the Report Viewing Options link in the ReportCast page banner, or enter a URL similar to the following:

`http://<web server>/acweb/<e.reporting server>/__requestUserPreferences`

For example, if your web server's name is Druid, and your e.Reporting Server also resides on Druid (running on port 81):

`http://druid:81/acweb/druid/__requestUserPreferences`

A page similar to the following image appears.



Choose whether to view reports in Dynamic HTML (DHTML) format, or as Actuate reports using the Actuate Live Report Extension (LRX). The default viewing format is DHTML. Your system administrator may have set an e.Reporting Server-wide default, as well.

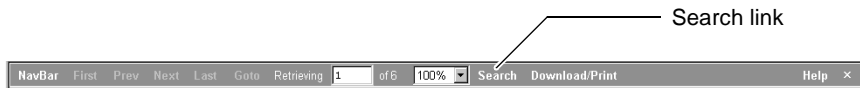
You can also download the Actuate Viewer LRX or change your password from the User Preferences page.

For more information about the User Preferences page, see *Using e.Reports*.

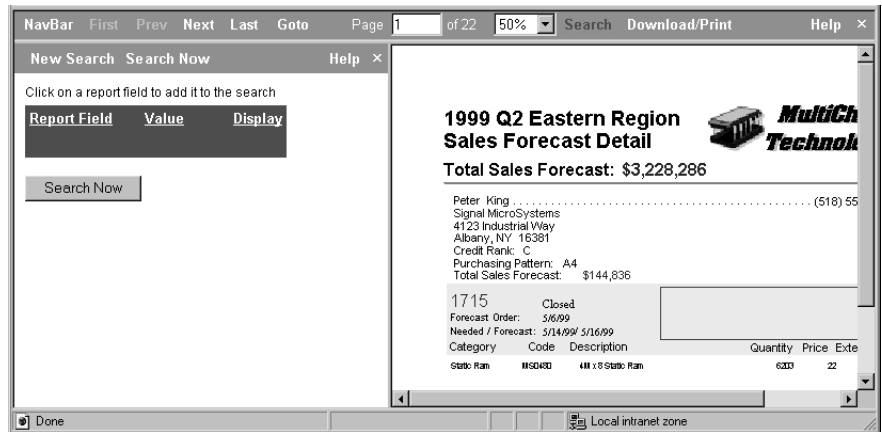
Searching DHTML reports

You can search for information while viewing DHTML reports using the Search Request page. To perform a search:

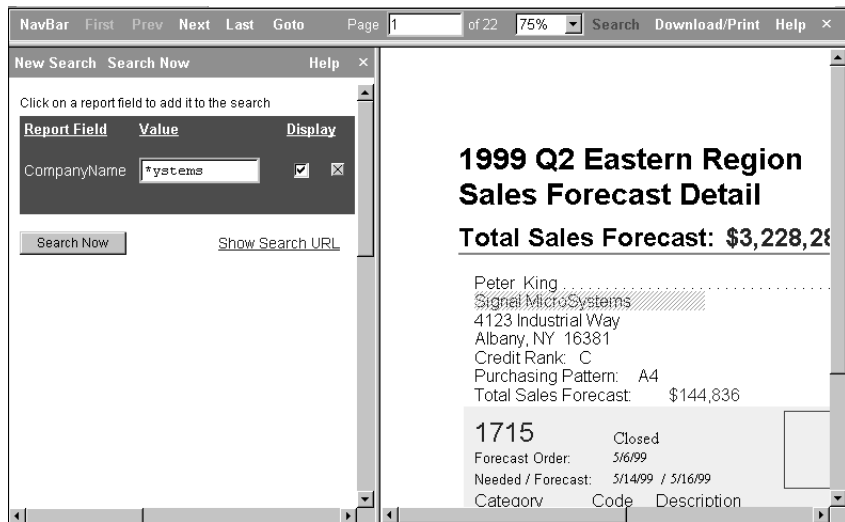
- 1 From the DHTML report viewing page, choose the Search link.



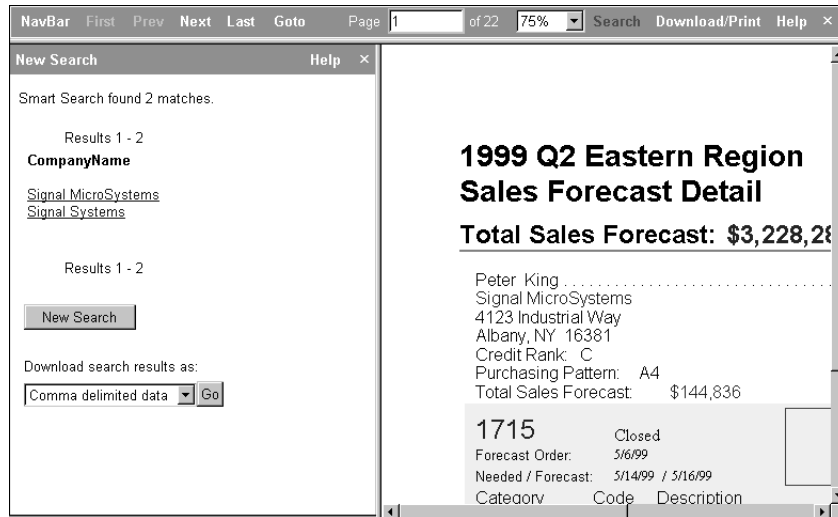
The Search Request page appears.



- 2 Select fields in the DHTML report to add them to your search criteria. The fields appear in the Search List.
- 3 Type the expression or value to search for. The Search Request page looks similar to the following image.



- 4 Click the Search Now button. The Report query results page appears in the browser.

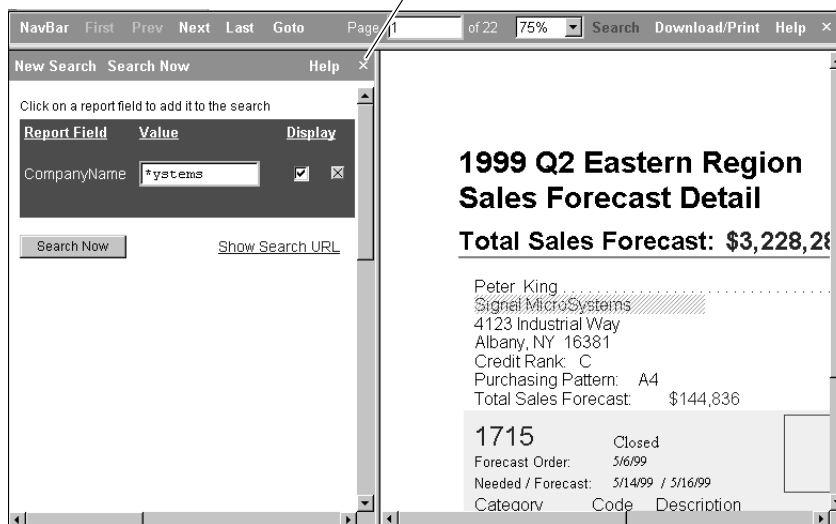


5 Click the search result links to examine specific search results.

To remove search criteria, you can do one of two things:

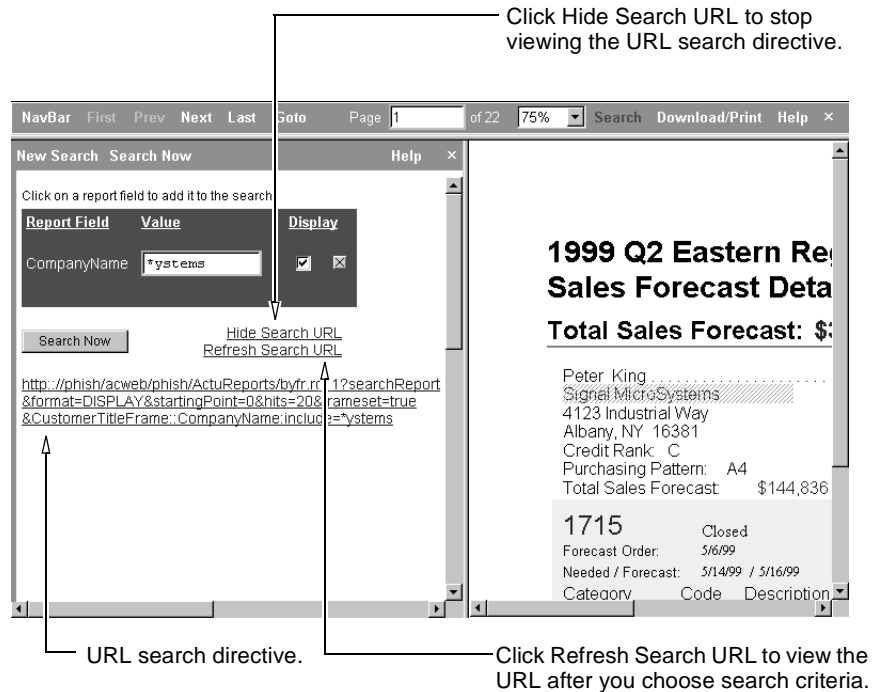
- Click a selected field to de-select it
- Click the X icon next to the selected field

Click the X icon to remove a search criteria.



Examining the search URL

You can examine the URL generated by your search. Click Show Search URL. A screen similar to the following image appears.



The search URL directive consists of:

- The report URL. The format is `http://<web server>/acweb/<report server>/<folder>/<filename>`. In this example, the report URL is:
`http://phish/acweb/phish/ActuReports/byfr.roi;1`
- The separator character, `?`. The separator character indicates that the next part of the URL is a directive.
- The search URL directive, `searchReport`.
- The search URL parameters.
- The search criteria, separated by `&`'s.

To stop viewing the search URL directive, click Hide Search URL. To see what the URL looks like after you add search criteria, click Refresh Search URL.

Click the URL link to activate the search, or drag it to another document to save it as a shortcut.

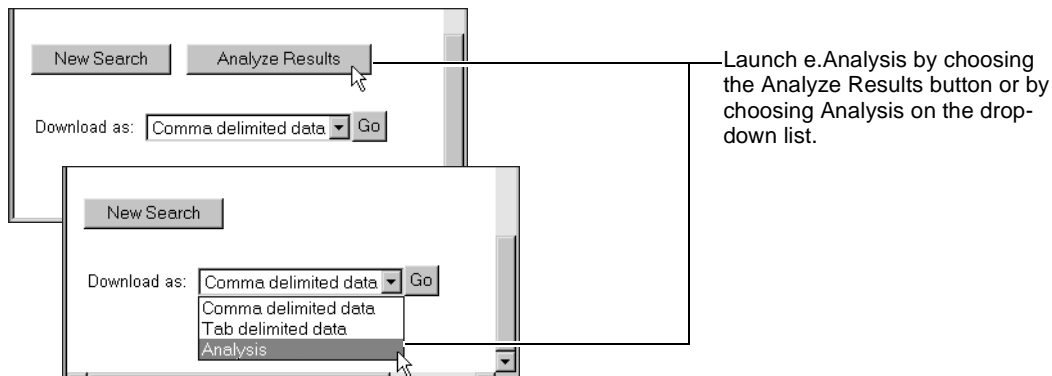
For more information about the search directive URL, see Chapter 4, “ReportCast directives,” later in this guide.

For more information about searching DHTML reports on the Web, see *Using e.Reports*.

Showing/Hiding the e.Analysis Button

The e.Analysis application is an add-on product that works with DHTML reports on the Web. You use e.Analysis to analyze data fields selected in a Smart Search. For more information, see *Using e.Analysis*.

If e.Analysis is installed, you launch the e.Analysis application by choosing the Analyze Results button or choosing Analysis on a drop-down menu in the Smart Search results pane, as shown in the following illustration.



Webmasters can customize how users launch e.Analysis. If e.Analysis is installed, check with your webmaster to find out how to launch e.Analysis.

Navigating through the Report Encyclopedia

You can navigate through the Report Encyclopedia in a way similar to navigating through the Administrator Desktop. From the main Report Encyclopedia web page, click on a folder's link. The folder contents page displays in the browser. You can continue clicking subfolder links to display their contents, click a report executable to generate a report, or click report .ROIs, .ROVs, or .ROWs to view reports.

Working with ReportCast channels

ReportCast channels provide easy access to particular reports of interest on the Web. When you subscribe to a channel, its link appears in your personal folder. Click the link to check the channel's contents and view reports as you wish.

Subscribing to a ReportCast channel

To subscribe to a ReportCast channel:

- 1 Choose the My Headlines tab. Your personal folder page appears.
- 2 Click the Personalize button. The channel subscription page appears.
The subscription page contains a list of available channels, with checkboxes to subscribe to them. Select the checkboxes for the channels to which you wish to subscribe. Click the Details icon to find out more about the channel.
- 3 Click the Apply button. A confirmation page similar to the subscription page displays in your browser.
- 4 Return to your personal channel. Reload the page. The channels you just subscribed to appear on your personal channel web page.

Unsubscribing from a ReportCast channel

To unsubscribe from a ReportCast channel:

- 1 Choose the My Headlines tab. Your personal folder page appears.
- 2 Click the Personalize button. The channel subscription page appears.
The subscription page contains a list of available channels, with checkboxes. A checked checkbox indicates that you are subscribed to a particular channel. Select the checkboxes for the channels you wish to unsubscribe from. The checkbox is unchecked.
- 3 Click the Apply button. A confirmation page similar to the subscription page displays in your browser.
- 4 Return to your personal channel. Reload the page. The channels you just unsubscribed from no longer appear on your personal channel web page.

Administering ReportCast

This chapter contains the following topics:

- About ReportCast administration
- Starting and stopping ReportCast
- Setting ReportCast default e.Reporting Server
- Changing the Windows servers ReportCast TCP/IP port
- Changing the ReportCast server machine
- Using non-default web server ports
- Configuring ReportCast port numbers
- Configuring the ReportCast Server host name
- Creating custom configurations
- Mapping custom configurations to locales
- Accessing UNIX ReportCast with CGI scripts
- Working with reports on Windows web sites
- Using CGI scripts for security
- Administering ReportCast channels
- URLs in e-mail notification
- Tuning ReportCast performance
- Viewing files using the native web server file system

About ReportCast administration

Administering ReportCast requires:

- Web server and report server administrator privileges
- Familiarity with web server and report server administration tasks
- Familiarity with ReportCast installations

ReportCast administration tasks include:

- Starting and stopping ReportCast
- Setting the default e.Reporting Server
- Changing the Windows server ReportCast TCP/IP port
- Changing the Windows server ReportCast Server machine
- Configuring communication port numbers
- Configuring the ReportCast Server host name
- Creating custom configurations
- Mapping configurations to system-defined language locales
- Accessing ReportCast using CGI scripts
- Working with reports on Windows sites
- Using CGI scripts for ReportCast security
- Administering ReportCast channels
- URLs in e-mail notification
- Tuning performance
- Viewing reports using the native web server file system

About ReportCast installations

ReportCast consists of the following components:

- ReportCast Agent handles the interface to the web server. ReportCast Agent is a dynamic-link library (DLL) that runs as part of your web server.
- ReportCast Server handles the interface to the Actuate e.Reporting Server. ReportCast Server is a service and runs as a process separate from the web server and ReportCast Agent.

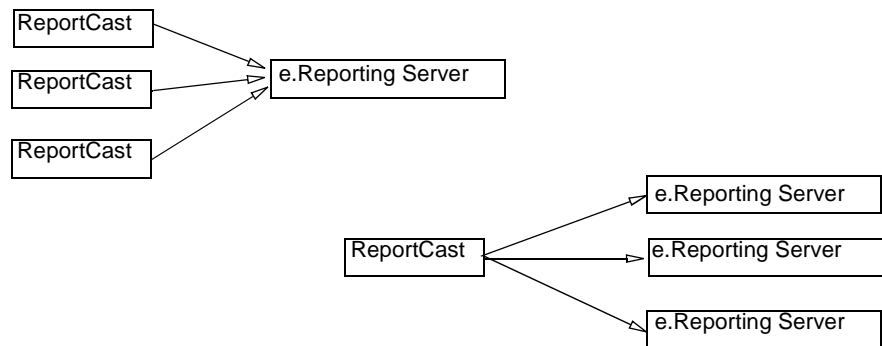
There is no user interface to ReportCast. There is no obvious result of installation.

Windows NT and Windows 2000 support one instance of a ReportCast Server per system. UNIX supports more than one instance of a ReportCast Server per system, as long as you configure each instance to listen on a different port. See Chapter 7, "Security for Internet deployment," for more information about configuring ReportCast Server and e.Reporting Server ports.

A single ReportCast Server can access multiple e.Reporting Servers. For example, you could use a single ReportCast Server accessing multiple e.Reporting Servers if you have a single web server hosting several applications, with each application using its own e.Reporting Server.

Multiple ReportCast servers can access a single e.Reporting Server. For example, you could use a multiple ReportCast Servers accessing one e.Reporting Server configuration if you have several web servers accessing the same e.Reporting Server resource.

The following diagram illustrates two possible ReportCast configurations.



Starting and stopping ReportCast

This section describes how to start and stop ReportCast on Windows servers and UNIX systems.

Starting and stopping ReportCast on Windows server systems

The Windows NT and Windows 2000 ReportCast Agent operates as part of the web server. ReportCast Agent starts and stops automatically when you start and stop the web server.

The Windows NT and Windows 2000 ReportCast Server is a service. To stop the Windows NT and Windows 2000 ReportCast Server service:

- 1 Start→Settings→Control Panel. The Control Panel window appears.
- 2 Double click Services. The Services box appears.
- 3 Click Actuate ReportCast Server 5.
- 4 Click the Stop button. The ReportCast Server service stops.

To start the Windows NT and Windows 2000 ReportCast Server service:

- 1 Start→Settings→Control Panel. The Control Panel window appears.
- 2 Double click Services. The Services box appears.
- 3 Click Actuate ReportCast Server 5.
- 4 Click the Start button. The ReportCast Server service starts.

Starting and stopping ReportCast on UNIX systems

UNIX ReportCast has these components:

- NSAPI extension or CGI component
- ReportCast service

The NSAPI or CGI components start automatically when the web server starts, or when a user navigates to a web reporting site URL. These components stop automatically when the web server stops.

The ReportCast service starts and stops separately.

Starting a UNIX ReportCast service

The ReportCast installation adds the ReportCast service startup script to the actuate/bin directory in the netscape/plugins directory for the web server that runs ReportCast. To start the ReportCast service, run the following script:

```
<web server directory>/startreportcast.sh
```

- <web server directory> is the netscape/plugins/actuate/bin directory path for the web server that runs ReportCast.
- startreportcast.sh is the ReportCast service startup script.

Stopping a UNIX ReportCast server

To stop the UNIX ReportCast server, run the following script:

```
<web server directory>/stopreportcast.sh
```

- <web server directory> is the netscape/plugins/actuate/bin directory path for the web server that runs ReportCast.
- stopreportcast.sh is the ReportCast server shutdown script.

Installation process for Sun Solaris and IBM-AIX

The ReportCast installation process for the Sun Solaris and IBM-AIX ReportCast service adds the service startup script to the Netscape web server startup script. Sun Solaris and IBM-AIX ReportCast services start when the web server starts.

Setting ReportCast default e.Reporting Server

You can define a default e.Reporting Server for ReportCast. This allows you to make use of the __ default e.Reporting Server specification in ReportCast directives. You use the registry key on Windows NT and Windows 2000 or environment variable on UNIX
AC_REPORTCAST_DEFAULT_REPORT_SERVER to set a default e.Reporting Server for ReportCast.

How to set the default e.Reporting Server on Windows servers

Use Regedit to create a new registry key on Windows NT and Windows 2000 systems. Before you begin, it is essential to back up your system.

- 1 From the Start menu, run Regedit.
The Registry Editor window displays.
- 2 Choose HKEY_LOCAL_MACHINE\Software\Actuate\Actuate ReportCast Server\5.
- 3 Select Edit.
The Edit menu appears.
- 4 Choose New→String Value.
A new value item displays in the right-hand side of the registry.
- 5 Type: AC_REPORTCAST_DEFAULT_REPORT_SERVER for the name of the new item.
- 6 Set the default e.Reporting Server:
 - 1 Select AC_REPORTCAST_DEFAULT_REPORT_SERVER.
 - 2 Right-click on AC_REPORTCAST_DEFAULT_REPORT_SERVER.

3 Choose Modify.

The Edit String dialog box appears.

4 Type the value of the string in the Value data field.

5 Choose OK.

For more information about specifying e.Reporting Servers in ReportCast directives, see “Formatting directives” in Chapter 4, “ReportCast directives.”

How to set the default e.Reporting Server on UNIX

1 Use a text editor to edit the startreportcast.sh script.

2 In the script, above the line that calls reportcastsrvr, add lines with the following syntax:

```
export AC_REPORTCAST_DEFAULT_REPORT_SERVER
AC_REPORTCAST_DEFAULT_REPORT_SERVER=<ReportServerName>
```

<ReportServerName> is the name of the default e.Reporting Server.

For example, to set ReportCast's default e.Reporting Server to an e.Reporting Server called Corporate, the lines you add read as follows:

```
export AC_REPORTCAST_DEFAULT_REPORT_SERVER
AC_REPORTCAST_DEFAULT_REPORT_SERVER=Corporate
```

For more information about specifying e.Reporting Servers in ReportCast directives, see “Formatting directives” in Chapter 4, “ReportCast directives.”

Changing the Windows servers ReportCast TCP/IP port

You can change the TCP/IP port that Windows NT and Windows 2000 ReportCast Agent and ReportCast server use to communicate. You must make the change on both the ReportCast server and ReportCast Agent machines. Use regedit to modify the registry key AC_REPORTCAST_PORT in HKEY_LOCAL_MACHINE/SOFTWARE/Actuate/Actuate ReportCast Server/5 on the ReportCast Server machine, and HKEY_LOCAL_MACHINE/SOFTWARE/Actuate/Actuate ReportCast Agent/5 on the ReportCast Agent machine. Before you begin, back up your system.

On both the ReportCast Agent and ReportCast server machines, perform the following steps to change the TCP/IP port:

- 1 From the Start menu, run Regedit.
- 2 The Registry Editor window displays.
- 3 Edit the AC_REPORTCAST_PORT registry keys:
 - 1 On the ReportCast server machine, choose HKEY_LOCAL_MACHINE\Software\Actuate\Actuate ReportCast Server\5.

On the ReportCast Agent machine, choose HKEY_LOCAL_MACHINE\SOFTWARE\Actuate\Actuate ReportCast Agent\5.
 - 2 Select AC_REPORTCAST_PORT.
 - 3 Select Edit.
 - 4 The Edit menu appears.
 - 5 Choose New→String Value.
 - 6 A new value item displays in the right-hand side of the registry.
 - 7 Type the new port number.
- 4 Verify that the value for AC_REPORTCAST_PORT on the ReportCast Agent machine matches the value of AC_REPORTCAST_PORT on the ReportCast server machine.
- 5 Choose OK.

When you are finished, restart the web server on the ReportCast Agent machine, and restart the ReportCast server on the ReportCast server machine.

Changing the ReportCast server machine

You can change the ReportCast server with which a ReportCast Agent communicates. You make the change on the ReportCast Agent machine. Use regedit to modify the registry key AC_REPORTCAST_HOST in HKEY_LOCAL_MACHINE\SOFTWARE\Actuate\Actuate ReportCast Agent\5. Before you begin, back up your system.

Perform the following steps to change the ReportCast server with which a ReportCast Agent communicates:

- 1 From the Start menu, run Regedit. The Registry Editor window appears.

- 2 Choose HKEY_LOCAL_MACHINE\Software\Actuate\Actuate ReportCast Agent\5.
- 3 Select AC_REPORTCAST_HOST.
- 4 Select Edit. The Edit menu appears.
- 5 Choose New→String Value. A new value item displays in the right-hand side of the registry.
- 6 Type the name of the machine that hosts the ReportCast server to communicate with.
- 7 Choose OK.

When you are finished, restart the web server on the ReportCast Agent machine.

Using non-default web server ports

The default port for most web servers is 80, or 8080. If your web server uses a non-default port, you must either specify the port as part of the URL, or set up an alias to redirect web browsers to another URL. The ReportCast installation process automatically installs ReportCast to communicate with the web server's port. However, when you set up links to your reporting web site, or define base URLs for accessing files on your reporting web site, specify the web server port.

For example:

`http://mustique:81/acweb/Sales`

indicates that the web server runs on a machine called Mustique, using port 81.

See your web server documentation for information about setting up web server aliases and redirecting web browsers.

Configuring ReportCast port numbers

You can specify a range of port numbers for ReportCast to use to connect to the e.Reporting Server. You might need to configure the port numbers if your web server uses a specific port number, such as a firewall.

On Windows NT and Windows 2000, you configure the start of the range of port numbers by setting a registry string value with the name AC_SERVER_SOCKET_BASE. The range of port numbers is set using a string

value with the name AC_SERVER_SOCKET_COUNT. The string value names are added to the key HKEY_LOCAL_MACHINE\Software\Actuate\Actuate Report Server\5. The values must be positive integers and the maximum value for both is 32767.

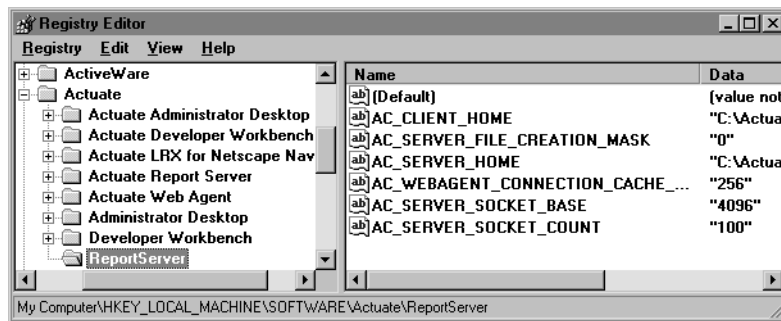
On UNIX, you set the AC_SERVER_SOCKET_BASE and AC_SERVER_SOCKET_COUNT environment variables.

If the start of the range of port numbers is not set, or the start of the range of port numbers is less than 1, the default value is 4096. If the count is not set, the count default value is 8192. If the count is set to less than 1, the count value defaults to 1.

How to configure ReportCast port range on Windows servers

- 1 If necessary, stop the e.Reporting Server, web server, and ReportCast Server processes. (ReportCast Agent is part of the web server processes.)
- 2 Start the registry editor REGEDIT.EXE.
- 3 Navigate to HKEY_LOCAL_MACHINE\Software\Actuate\Actuate Report Server\5.
- 4 Create the string value name AC_SERVER_SOCKET_BASE and the value of first port number.
- 5 Create the string value name AC_SERVER_SOCKET_COUNT and the value of the range of port numbers.

This example shows the base port number set to 4096 and the range of port numbers set to 100.



- 6 Close the registry editor.
- 7 Restart the e.Reporting Server and web server processes.

For more information about how to set registry keys, see "How to set a new locale registry key on a Windows server system," later in this chapter.

How to configure ReportCast port range on UNIX

- 1 If necessary, stop the e.Reporting Server and web server processes. ReportCast is part of the web server processes.
- 2 Add lines to set the environment variables to e.Reporting Server request server startup script `$AC_SERVER_HOME/bin/reqsrvr.sh`.

Add the new line before the first line that uses the export command.

For example, add these lines to change the `AC_SERVER_SOCKET` base port number to 4096 and the range to 100:

```
AC_SERVER_SOCKET_BASE=4096
export AC_SERVER_SOCKET_BASE
AC_SERVER_SOCKET_COUNT=100
export AC_SERVER_SOCKET_COUNT
```

- 3 Start the e.Reporting Server.

For more information about how to set environment variables, see “How to set locale environment variable on a UNIX system,” later in this chapter.

Creating custom configurations

ReportCast comes with standard configurations such as `enu` (U.S. English) and `jpn` (Japanese). The default configuration is called `default`. To work with custom configurations, you must add `<config>` to your URL. If you do not add `<config>`, the ReportCast uses `default` as the configuration.

The URL has the following syntax:

```
http://<webserver>/acweb/<config>/<reportserver>
```

- `<webserver>` is your web server name.
- `acweb` is a literal string indicating a directive for ReportCast.
- `<config>` is the name of the configuration to use.
- `<reportserver>` is the name of the e.Reporting Server to access.

How to create a custom configuration

You can create custom configurations, for example, for a particular division or department:

- 1 Go to the web server’s `Actuate` directory.
- 2 Add a new directory named for the custom configuration.
- 3 In the custom configuration directory, add a directory named `Standard`.

- 4 Copy the standard ReportCast templates to the custom configuration's Standard directory.

How to set up directories for the custom templates in a custom configuration

- 1 In the custom configuration directory, create a directory called Custom.
- 2 Copy the templates you want to customize into the Custom directory.
- 3 Customize the templates as appropriate.

For more information about customizing templates, see Chapter 5, "ReportCast scripting language reference."

Mapping custom configurations to locales

ReportCast maps the name of its standard configurations to localization settings, also known as locales, to determine how to print numbers and dates and how to sort strings. The differences are significant to users in those language groups. For example, the French, English, and Japanese locales display date and time values differently.

The names of configurations and locales are usually the same.

Locales are standard operating system features. They have standard three-letter names, such as `enu`, `jpn`, and `frn`, for U.S. English, Japanese, and French. When you create a custom configuration that does not have the same name as its locale, you map the custom configuration to the appropriate locale to enable the appropriate functionality. On a Windows NT or Windows 2000 system, you create a new registry key on the machine which hosts the ReportCast Server. On a UNIX system, you set an environment variable on the machine which hosts the ReportCast Server.

Folder items and other lists sorted by string values display in order according to the locale of the machine running ReportCast, not necessarily according to the locale specified by the configuration in the URL.

How to set a new locale registry key on a Windows server system

Use Regedit32 to create a new registry key on the Windows NT or Windows 2000 system that hosts the ReportCast Server. Before you begin, it is essential to back up your system.

- 1 From the Start menu, run Regedit.

The Registry Editor window displays.

2 Choose HKEY_LOCAL_MACHINE\Software\Actuate\Actuate ReportCast Server\5.

3 Choose Edit.

The Edit menu appears:

1 Choose Add Value.

2 The Add Value dialog displays.

3 Type: AC_REPORTCAST_<config>_LOCALE for the name of the new item.

4 <config>, the name of the custom configuration, must be typed in uppercase letters.

5 Set the datatype as REG_SZ.

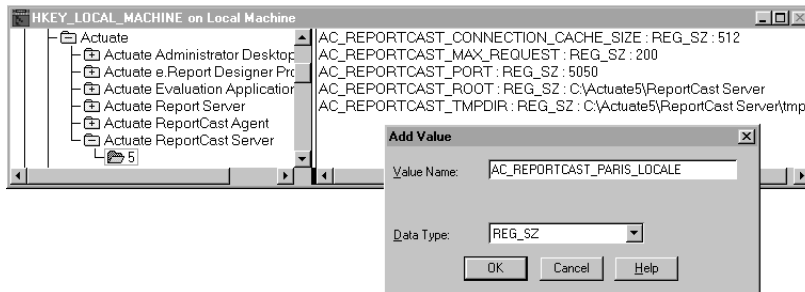
6 Choose OK. The String Editor dialog appears.

7 Type the locale in the String field. For example, if the locale is French, type frn. If the locale is U.S. English, type enu.

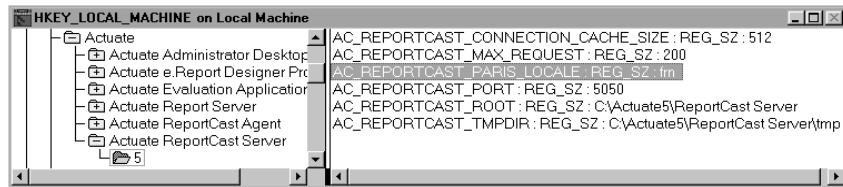
4 Click OK.

5 Close Regedit32.

For example, if you have a custom configuration for your company's French office called PARIS, and you want the configuration to use the frn locale information, set AC_REPORTCAST_PARIS_LOCALE to frn. The following illustration shows you how to modify the registry key.



The following illustration shows the modified registry key.



How to set locale environment variable on a UNIX system

- 1 Use a text editor to edit the startreportcast.sh script.
- 2 In the script, above the line that calls reportcastsrvr, add lines with the following syntax:

```
export AC_REPORTCAST_<config>_LOCALE  
AC_REPORTCAST_<config>_LOCALE=<LocaleName>
```

<config> is the name of the custom configuration.

<LocaleName> is the name of the locale.

For example, to set the BUREAUFRANCE locale on your UNIX system, the lines you add read as follows:

```
export AC_REPORTCAST_BUREAUFRANCE_LOCALE  
AC_REPORTCAST_BUREAUFRANCE_LOCALE=frn
```

Supporting locales with e.Reporting Server and ReportCast

The Actuate e.Reporting Server uses a localemap file to support different locales when converting reports in DHTML, XML, or PDF formats. For more information about localemap file, see *Actuate e.Reporting Server Guide*.

ReportCast also uses a localemap file to support different locales. The localemap file is located in the web server's document root directory, for example, <webroot>\docs\actuate\etc.

Any modifications made to the localemap file on an e.Reporting Server system must also be made on any ReportCast system that connects to the e.Reporting Server. The same system locales on the e.Reporting Server system must also be installed on the ReportCast system to allow Actuate locale support to work with Actuate reports in DHTML, XML, and PDF formats.

Accessing UNIX ReportCast with CGI scripts

If your UNIX system uses CGI scripts to access ReportCast, consider the following issues:

- If the ReportCast service is not running and a user attempts to connect to ReportCast, the CGI implementation returns an error. The NSAPI implementation returns a blank page.
- If your web server automatically generates response headers, such as cookies, consider using a parsed-header CGI script to access ReportCast.

- CGI scripts do not automatically resolve escape sequences into the special characters they represent. Your CGI script must include code that translates escape sequences. An escape sequence begins with the percent character (%), followed by the ASCII two-digit hexadecimal code for the special character. For example, add code to translate the following URL:

`http://RS%22Main/acweb/SalesRpts`

to:

`http://RS_Main/acweb/SalesRpts`

CGI scripts are supported only with UNIX ReportCast. There is no CGI counterpart supported for Windows NT and Windows 2000.

How to use a parsed-header CGI script

- 1 Export the environment variable:
`HTTP_AC_REPORTCAST_PARSED_HEADERS`
- 2 In the CGI script, set the environment variable to True.

The ReportCast installation process generates two CGI scripts:

- `nph-actuate.cgi`
- `ph-actuate.cgi`

The script `ph-actuate.cgi` is a parsed-header script. A URL that contains this script name generates output to which the web server can add its own headers.

If you use a parsed-header CGI script to access ReportCast, your URL must reference `ph-actuate.cgi` to produce parsed-header output.

Working with reports on Windows web sites

To work with reports on a Windows-based web reporting site, register a browser plug-in or helper application for the MIME types of Report Encyclopedia items. You can register plug-ins and helper applications in your browser. For example, in the Netscape browser, register a helper application from Options→General Preferences→Helpers.

If no browser plug-in or helper application is registered for a requested Report Encyclopedia item, the web browser prompts you to save the item to the local file system. The default file name provided in the Save dialog box includes a version number, for example, `REPORT.ROI;1`. Windows systems use the filename extension to recognize the file type. To ensure that Windows can recognize the file type, delete the version part of the filename, including the semicolon. In this example, you save the file as `REPORT.ROI` because

Windows does not accept the ;1 convention. You can also modify the name to something like REPORT1.ROI to add versioning information.

If the name of the Report Encyclopedia item contains spaces or other special characters, the default file name contains character sequences beginning with % instead of the actual characters.

Using CGI scripts for security

You may need to mask the location on the report server of reports that a user views on the web reporting site.

For example, a custom HTML page can display a link to a report at:

`http://webserver/cgi/secure.cgi/myreport`

instead of the actual location. For example:

`http://webserver/cgi/secure.cgi/acweb/reportserver/accounts/trusted/
myreport.roi?View`

The CGI script masks the report URL on the web reporting site. To mask the report URLs and use the CGI script to reconstruct the complete URLs:

- Set the CGI script to change the incoming, disguised values of the PATH_INFO and QUERY_STRING CGI variables to construct a valid report-viewing request URL.
- Set the CGI script to set and export the environment variable HTTP_AC_REPORTCAST_BASE_URL to the unmasked URL.

The Base URL is that part of the URL up to and including the name of the machine running the report server through which the report is normally accessed on the web reporting site.

In this example:

- The incoming value of PATH_INFO is /myreport. The CGI script resets it to /acweb/reportserver/accounts/trusted/myreport.roi.
- The incoming value of QUERY_STRING is empty. The CGI script resets it to View.
- The Base URL value is /cgi/secure.cgi/acweb/reportserver, or, alternatively, http://webserver/cgi/secure.cgi/acweb/reportserver.

Administering ReportCast channels

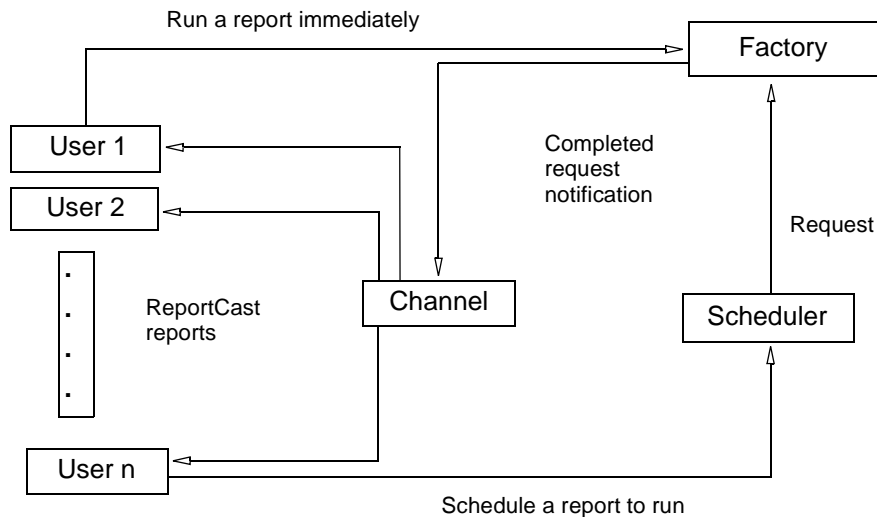
This section describes how to administer ReportCast channels.

About Actuate ReportCast channels

Actuate ReportCast channels is a service that allows users to subscribe to particular web channels and view reports available on those channels. ReportCast channels uses push technology to deliver reports to particular web channels. Push technology is also known as web casting, push distribution, or publish and subscribe. When new reports become available, the appropriate channel receives a notification. Subscribed users check the channel's web page to get a listing of available reports. When users click on a report link, the report appears in the web browser window.

ReportCast channels allows you to subscribe to and unsubscribe from particular channels, list the channels to which you are currently subscribed, and view the contents of those channels. If you have administrator privileges, you can also create, delete, and modify channels.

The following illustration shows how ReportCast channels fit into the Actuate Internet architecture.



Users make requests to run reports immediately, or to run reports at a scheduled time. The requester form allows users to select the users and channels to notify when the report is complete. Once the report runs successfully, the completed request notification goes to the appropriate users and channels.

For more information about factory processing, see Chapter 8, “Understanding the report generation process,” in *Programming e.Reports*. For more information about scheduling reports, see Chapter 4, “Running a report from the desktop,” in *Using e.Reports*.

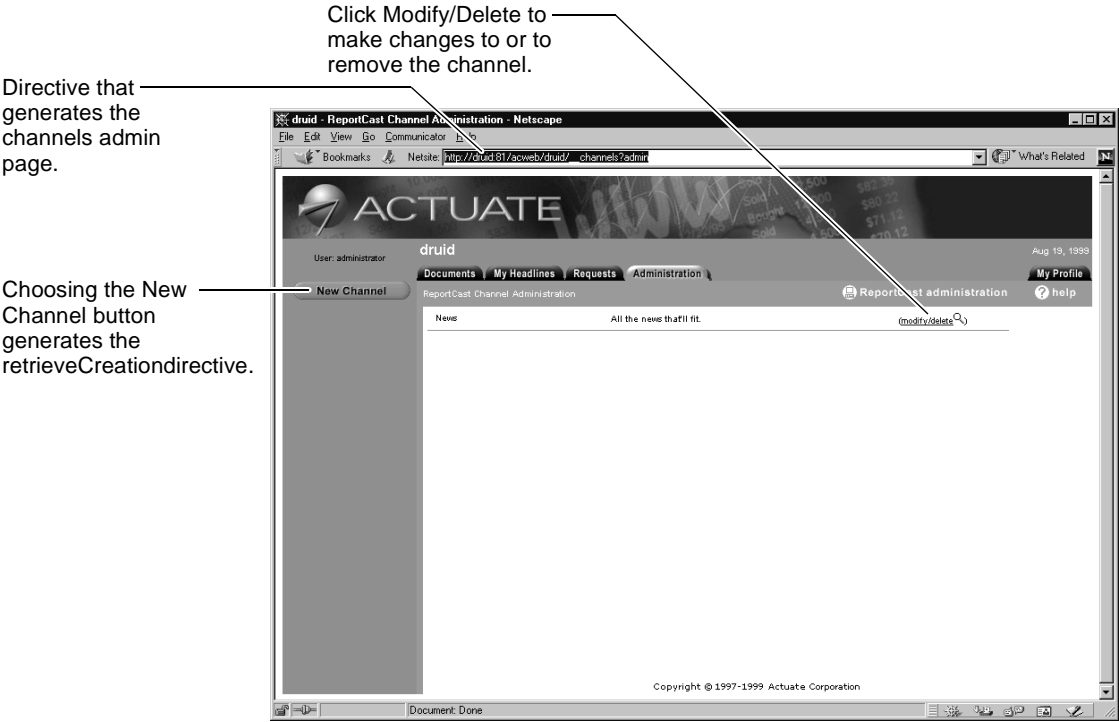
Users with administrative privileges on e.Reporting Server can administer ReportCast channels. Choose the Administration tab on the Report Encyclopedia web page, or use the `__channels?admin` directive to access the channel administration page. The complete syntax for the directive is:

`http://<webserver>/acweb/<reportserver>/__channels?admin`

The following table lists the ReportCast channels administrative directives. All commands are available from the channels administration page.

Command	Description
<code>__channels?admin</code>	Accesses the ReportCast channels administration web page.
<code>__completed</code>	Lists the channels to which this user subscribes.
<code>__channels/<channel></code>	Lists the contents of channel <code><channel></code> .
<code>__channels?retrieveForm</code>	Retrieves and displays the channels subscription form.
<code>__channels?submitForm</code>	Submits the subscription form to ReportCast.
<code>__channels/<channel>?drop</code>	Removes the specified channel from the e.Reporting Server.
<code>__channels/<channel>?retrieveDetails</code>	Displays the details form for the specified channel. Users with administrator privileges can modify the channel using this form.
<code>__channels/<channel>?submitDetails</code>	Submits the form that modifies details of a channel.
<code>__channels?retrieveCreation</code>	Displays the form that allows users with administrator privileges to create a new channel.
<code>__channels?submitCreation</code>	Submits the form that creates a new channel.

The following illustration shows the channels administration page.



When you click the New Channel button, the Channel Administration page appears.

The screenshot shows the ACTUATE Channel Administration page. The page has a header with the ACTUATE logo and a navigation bar with tabs: Documents, My Headlines, Requests, and Administration. The Administration tab is selected, and the sub-tab is 'ReportCast Channel Administration > _newChannel - Detail'. The page is titled 'druid' and shows the user 'administrator' and the date 'Sep 06, 1999'. There is a 'My Profile' link and a 'help' icon.

Annotations point to the following elements:

- Click Apply to create the new channel.** Points to the 'Apply' button.
- Enter the URL for the scaled-down image to use for this channel's icon.** Points to the 'Small icon URL' input field.
- Enter the URL for the large image to use for this channel's icon.** Points to the 'Large icon URL' input field.
- Select a Security checkbox to set the privileges needed to access the new channel.** Points to the 'All' and 'Manager' checkboxes under the 'Security' section.

The form fields include:

- Name:** A text input field.
- Description:** A large text area.
- Small icon:** A small satellite dish icon.
- Small icon URL:** A text input field with dimensions (width=20, height=20).
- Large icon:** A large satellite dish icon.
- Large icon URL:** A text input field with dimensions (width=32, height=32).
- Refresh Every:** A text input field with '180' and 'seconds'.
- Auto-delete:** A text input field with '14' and 'days'.
- Security:** A section with the text 'Specify roles that have access to this channel.' and two checkboxes: 'All' and 'Manager'.

Copyright © 1997-1999 Actuate Corporation

How to create a new channel

To create a new channel from the channels administration page:

- 1 Choose New Channel.**
The new channel creation page displays in the browser.
- 2 Fill out the channel creation form:**
 - 1** Name is the name of the channel as it appears in channel subscription lists.
 - 2** Description is text describing the channel's purpose, audience, or other pertinent information.
 - 3** Small Icon URL is the URL to an image, for example, .GIF, or .JPG, to be used as the small-scale icon for this channel. The default is the satellite dish icon.

- 4 Large Icon URL is the URL to an image, for example, .GIF, or .JPG, to be used as the large icon for this channel. The default is the satellite dish icon.
 - 5 Refresh Every determines how often the channel's contents should be updated.
 - 6 Auto-delete determines how long an item may remain in the channel before it is automatically deleted.
 - 7 Use the security section to determine which roles may access the channel.
- 3 Click Apply to create the channel.
- A confirmation page displays in the browser, showing the new channel's name and details.

How to modify an existing channel

To modify an existing channel from the channel's administration page:

- 1 Click the Modify/Delete icon for the channel you wish to modify.
The channel's Details page displays in the browser.
- 2 Edit the channel's information on the Details page.
- 3 Click Apply.
The updated channel's Details page displays in the browser.

How to delete an existing channel

To delete an existing channel from the channel's administration page:

- 1 Click the Modify/Delete icon for the channel you wish to modify.
The channel's Details page displays in the browser.
- 2 Click Delete Channel.
The Successful command completion page displays in the browser.

URLs in e-mail notification

Report Encyclopedia administrators can configure the e.Reporting Server to send an e-mail notification when a request is complete. Using ROTP syntax to specify the report's location is the default. ROTP syntax is a report object protocol that runs a report on a server. For example, an e-mail sent to users that contains the line:

```
rotp://bimini/Sales/goals/quarterly.roi;7
```

specifies the seventh version of the report QUARTERLY.ROI is in the folder Sales\Goals in the Report Encyclopedia on Bimini.

As administrator, you can configure the e.Reporting Server to specify a URL for the location of the report when notifying users by e-mail. For example, if users can access the Report Encyclopedia using ReportCast on the web server Paradise, you can configure the e.Reporting Server to specify:

```
http://paradise/acweb/bimini/Sales/goals/quarterly.roi;7?view
```

For more information about specifying URLs, see Chapter 6, "Administering the Report Encyclopedia in the web environment," in *Administering the Report Encyclopedia*.

Tuning ReportCast performance

This section describes some considerations and options for tuning ReportCast performance.

Setting user connection timeouts

Each time a user logs on to the web reporting site, ReportCast creates a connection between itself and the e.Reporting Server. The connection stays open indefinitely unless you specify a timeout period.

Registry keys on Windows NT and Windows 2000, or environment variables on UNIX, allow you to set the timeout period:

- **AC_REPORTCAST_CONNECTION_CACHE_SIZE** sets the maximum number of e.Reporting Server connections that ReportCast holds open at any time. Set **AC_REPORTCAST_CONNECTION_CACHE_SIZE** to the number of actual users that will be accessing ReportCast, not necessarily the number of distinct e.Reporting Server users. If the connection cache size is not set, the number of connections is unlimited. Set **AC_REPORTCAST_CONNECTION_CACHE_SIZE** to 0 (zero) to ensure

that connections are never cached from one request to another. The default value is 256.

- `AC_REPORTCAST_CONNECTION_CACHE_TIMEOUT` sets the approximate time, in seconds, after which an idle e.Reporting Server connection is dropped. If connection cache timeout is not set, connections remain on the e.Reporting Server indefinitely.

On Windows NT and Windows 2000, set the registry keys as string values in `HKEY_LOCAL_MACHINE\Software\Actuate\Actuate Web Agent\5.0`. Use Regedit to set the registry keys. For more information about how to set registry keys, see “How to set a new locale registry key on a Windows server system,” and “How to set locale environment variable on a UNIX system,” earlier in this chapter.

Setting request timeouts

You can set the length of time that ReportCast waits before a request times out. The `AC_REPORTCAST_REQUEST_TIMEOUT` registry key on Windows NT and Windows 2000 or environment variable on UNIX specifies the request timeout period in seconds. It affects only those report requests for which users checked the Wait checkbox (wait for completion). If `AC_REPORTCAST_REQUEST_TIMEOUT` is set and the Wait checkbox is checked on the requester form, ReportCast waits the specified number of seconds. If the report request has not completed, ReportCast displays the request status page, rather than waiting for the request to complete.

The default is that there is no timeout.

On Windows NT and Windows 2000, set the registry keys as string values in `HKEY_LOCAL_MACHINE\Software\Actuate\Actuate Web Agent\5.0`. Use Regedit to set the registry keys. For more information about how to set registry keys, see “How to set a new locale registry key on a Windows server system,” and “How to set locale environment variable on a UNIX system,” earlier in this chapter.

Tuning viewing performance

If your users are waiting too long for reports to display over the net, set the `DemandCachePageSize` registry key to tune performance.

`DemandCachePageSize` controls the size of data packets sent to an Actuate Viewer LRX over the network. Specify the data packet size in bytes. Packet size must be a multiple of 2048:

- Minimum packet size is 2048 bytes
- Maximum packet size is 128K bytes
- Default packet size is 64K

Consider the following items when deciding how to set the DemandCachePageSize registry key:

- Smaller packet sizes cause more reads to be made to the network. If you have a fast network, larger packet sizes may work well for your users.
- Larger packet sizes cause fewer reads to be made to the network. In general, read time is shorter than connect time.
- You may need to experiment to find the packet size that works best for your users. Variables such as network speed, number of users, and so on affect how much particular packet sizes increase report viewing efficiency for your users.

How to set the DemandCachePageSize registry key

For more information about using Regedit, see “How to set a new locale registry key on a Windows server system,” and “How to set locale environment variable on a UNIX system,” earlier in this chapter.

- 1 Run Regedit:
 - For the Netscape browser, navigate to HKEY_CURRENT_USER→Software→Actuate→Netscape LRX 4.0→General
 - For the Microsoft Internet Explorer browser, navigate to HKEY_CURRENT_USER→Software→Actuate→MSIE LRX 4.0→General
- 2 Select DemandCachePageSize.
- 3 Set the value of DemandCachePageSize.

Viewing files using the native web server file system

Some web server-based viewing technologies expect to be able to directly access a file by reading it from the web server’s file system. You can store non-Actuate objects in the Report Encyclopedia and view them on the Web using their intended viewing technology.

- 1 In the Administrator Desktop, enable the Use Web Server File System feature. See Administering the Report Encyclopedia for information about enabling the Use Web Server File System feature.
- 2 Ensure that the viewing technology is installed on the machine hosting the ReportCast Agent.
- 3 Use your web browser to access the e.Reporting web site.

- 4 View a report of the file type for which you enabled the Use Web Server File System feature. Users notice only that the report appears in its native viewer, rather than in Actuate's viewer or DHTML formats.

ReportCast copies the report to a temporary directory on the web server's file system for viewing. The temporary directory, ViewTemp, resides in the Actuate directory in the web server's document root directory. ReportCast disables automatic indexing for the ViewTemp directory for the default installation.

ReportCast generates random, unique file names for the reports. The names have no meaning in relation to the reports. For example, V1_f49_f99808e4_sgth8ft3_1c0-93-5.rpt.

Configuring viewing options

ReportCast administrators configure viewing options by setting environment variables on UNIX systems, or registry keys on Windows NT and Windows 2000 systems.

The following table lists and describes the configuration options for viewing reports on the web server's native file system.

Native web server file system viewing option	Description
AC_REPORTCAST_VIEWTMPDIR	The full pathname of the temporary directory. The directory must exist in the Actuate directory of the web server document root directory structure.
AC_REPORTCAST_VIEWTMPDIR_VIRTUAL	The location of the temporary directory relative to the document root of the web server. ReportCast uses this value to hide the actual location of the temporary directory from users. AC_REPORTCAST_VIEWTMPDIR_VIRTUAL indicates how to get to the directory from a web browser. This path can be completely different from the physical location. ReportCast Agent uses this value to redirect the browser for viewing the report.
AC_REPORTCAST_VIEWTMPDIR_MAX_SIZE	The maximum allowed size of the temporary directory, in kilobytes. When this value is not set, there is no limit.
AC_REPORTCAST_VIEWTMPDIR_MAX_FILE	The maximum number of files allowed in the temporary directory. When this value is not set, there is no limit.

Native web server file system viewing option	Description
AC_REPORTCAST_VIEWTMPDIR_FILE_PERSISTENCY	The length of time the file remains in the temporary directory, in seconds. The default is 900, or 15 minutes. The minimum value is 60.
AC_REPORTCAST_VIEWTMPDIR_CLEANING_FREQUENCY	The frequency with which the automatic cleaning process removes files from the temporary directory, in seconds. The default value is 30.

Files remain in the temporary directory between the value of AC_REPORTCAST_VIEWTMPDIR_FILE_PERSISTENCY and the value AC_REPORTCAST_VIEWTMPDIR_FILE_PERSISTENCY + AC_REPORTCAST_VIEWTMPDIR_CLEANING_FREQUENCY. The value of AC_REPORTCAST_VIEWTMPDIR_FILE_PERSISTENCY must be higher than AC_REPORTCAST_VIEWTMPDIR_CLEANING_FREQUENCY.

Handling viewing security issues

Report Encyclopedia security does not apply while the report is on the web server's hard drive. To help prevent unauthorized access of reports, take the following precautions:

- Restrict access to the temporary directory, or restrict logon to the web server host machine.
- Ensure that the directory listing is disabled, to prevent users who know the name of the directory from accessing it over the Web.
- Users guessing the temporary file names is an unlikely scenario as the temporary file names are not obvious.

Working with ReportCast templates

This chapter contains the following topics:

- About ReportCast templates
- Creating custom templates

About ReportCast templates

As you submit directives, ReportCast generates reporting web pages dynamically from HTML templates. The template files contain scripting language commands embedded in HTML comments. The scripting language commands tell ReportCast how to generate and format its web pages.

You can customize the templates to meet your needs. Only the downloading and viewing directives cannot use custom templates. You can also customize default forms, perhaps modifying a standard form to display only some of the available parameters, add a background color or graphic, or to display your company logo.

Actuate provides a set of default, standard templates with the ReportCast Server installation.

To customize ReportCast templates, you should have knowledge of the following:

- Directives you can send to ReportCast. For information about ReportCast directives, see Chapter 4, “ReportCast directives.”
- Contents of the template files. For information about template files, see “About HTML template files,” later in this chapter.
- ReportCast scripting language. For information about ReportCast scripting language, see Chapter 5, “ReportCast scripting language reference.”
- HTML, the HyperText Markup Language. Information about HTML is available from various sites on the World Wide Web and in many books available at local bookstores.
- JavaScript. Information about JavaScript is available from various sites on the World Wide Web and in many books available at local bookstores.

About HTML template files

All ReportCast template files must have the extension .ACHTML, for Actuate template HTML. Template files can be specific to display the contents of a particular folder on a particular report server. Template files can also be general to display e.Reporting Server objects on a particular web server. Whether a template applies to one e.Reporting Server folder or to the entire web server depends on its placement in the e.Reporting Server, the web server, or the ReportCast Server directory structure.

Custom HTML requester forms are the exception. These are HTML files that you create to run specific reports. ReportCast does not provide these forms. Custom HTML requester forms must:

- Reside in the same e.Reporting Server folder as the report executable (.ROX) file

- Have the file extension .HTML
- Depend upon the .ROX file

Custom HTML requester forms that reside on the e.Reporting Server in a Report Encyclopedia folder are static. Since custom HTML requester forms reside in the Report Encyclopedia, ReportCast cannot fill in variables such as the e.Reporting Server name and the current date and time.

For more information about the placement of template files, see “About template placement and scope,” later in this chapter.

For more information about creating custom HTML requester forms for specific reports, see “Creating custom requester forms for a specific report,” later in this chapter.

The following table lists and describes the web pages that ReportCast generates.

Generated HTML page	Template file name	Description
Folder Listing	FOLDERLIST.ACHTML	Content of an Encyclopedia folder.
Requester Menu	REQUESTERMENU.ACHTML	Request form that runs reports.
Standard Requester	REQUEST.ACHTML	Standard request form.
Custom Folder	INDEX.ACHTML	Custom folder page created by an administrator and stored directly on an e.Reporting Server.
Request Folder	REQUESTLIST.ACHTML	Content of one or all request folders.
Request Confirmation	CONFIRM.ACHTML	Response page returned after a user submits a request to run a report.
Request Detail	REQUESTDETAIL.ACHTML	Response page used to display detailed information about a request.
Request Status	STATUS.ACHTML	Status page returned after a user asks for request status.
Cancelled Request Confirmation	CANCEL.ACHTML	Response page returned when a request to cancel an active request is successful.

Generated HTML page	Template file name	Description
File Detail	FILEDETAIL.ACHTML	Lists file properties.
Confirmation	SUCCESS.ACHTML	Response page returned when a request to drop or rename a report is successful.
Error	ERROR.ACHTML	Response page generated after an error condition is detected.
Administration	ADMIN.ACHTML	Page generated to perform general, non-e.Reporting Server-related, administrative tasks.
Channel Subscription	CHANNELSUBSCRIPTION.ACHTML	Subscription form to subscribe to and unsubscribe from channels.
Channel Administration	CHANNELADMIN.ACHTML	Top-level channel administration page.
Channel Contents	CHANNELCONTENTS.ACHTML	Page that displays contents of a specified channel.
Channel Detail	CHANNELDETAIL.ACHTML	Administrators use this page to create, delete, and modify a specified channel.
Report Viewing Options	USERPREFERENCES.ACHTML	Choose how to view reports.
View Frameset	VIEWFRAMESET.ACHTML	The initial set of frames that display the DHTML report, including the navigation toolbar, search window, table of contents, and the DHTML report.
Navigation	VIEWNAV.ACHTML	The navigation toolbar used for viewing the DHTML report.
Table of Contents	VIEWTOC.ACHTML	The DHTML report's table of contents.
File Versions	FILEVERSIONS	Display the list of available versions of the current file.

About internal template structure

The standard ReportCast pages include many elements that several templates use, such as the copyright section and page headers. The ReportCast templates directory contains files with names such as `contentactive.achtml`, `contentbegin.achtml`, `contentfileversions.achtml`. These subtemplate files generate elements common to many ReportCast pages. ReportCast templates include these subtemplate files, using ReportCast scripting language include statements, to provide elements such as the copyright notice, page headers, and folder contents listing. To create custom templates, you customize the individual subtemplate files as well as the standard template files.

About template placement and scope

A template's placement determines its scope. A template can apply to a specific report, report server folder, or to an entire web server.

You can place templates in the Report Encyclopedia, or on the web server. Templates placed in the Report Encyclopedia do not have access to the ReportCast variables, such as the current date and time. However, placing templates in the Report Encyclopedia allows you to customize specific folders and create requester forms for specific reports.

Placement in an e.Reporting Server folder for a specific report

Placing an HTML requester form in an e.Reporting Server folder and creating a dependency from that requester form to a particular .ROX file provides a custom requester form for that specific report executable. If more than one HTML file in the e.Reporting Server folder depends upon a particular .ROX, ReportCast displays a requester menu form. The user chooses the custom requester to use.

Placement in an e.Reporting Server folder

Placing a template called `INDEX.ACHTML` in an e.Reporting Server folder provides a folder-specific template. Such templates apply only to a particular folder's contents on a particular e.Reporting Server.

Placement in an e.Reporting Server's __WebAgent folder

Creating a `__WebAgent` folder on an e.Reporting Server, then placing ReportCast templates inside, provides e.Reporting Server-specific templates. Such templates apply only to objects on a particular e.Reporting Server.

Placement in the Custom directory

Placing ReportCast templates in a directory called `Custom` in the web server's Actuate directory tree, or in the ReportCast Server's Actuate directory tree,

provides web server configuration-specific templates. Such templates apply only to objects for a particular configuration of a particular web server, for example, the Japanese-language or French-language configuration.

For installations with both ReportCast Agent and ReportCast Server on the same machine, locate the Custom directory in your web server's document root directory, in the Actuate directory. For installations with separate ReportCast Agent and ReportCast Server installations, locate the Custom directory in the Actuate distribution's ReportCast directory:

```
wwwroot\actuate\<config>\standard
```

where <config> is the name of the configuration. For example, if your web server configuration is on wwwroot, and you are providing a Japanese-language configuration, the directory path is:

```
wwwroot\actuate\jpn\custom
```

If your ReportCast Server is in the Actuate distribution directory /usr/actuate and you are providing a Japanese-language configuration, the directory path is:

```
/usr/actuate/ReportCast/templates/jpn/custom
```

If you plan to provide custom templates for a particular configuration, you must create the Custom directory yourself and place the custom templates in it.

Placement in the Standard directory

ReportCast templates located in a directory called Standard in the web server's Actuate directory tree, or in the ReportCast Server's Actuate directory tree, are the default ReportCast templates supplied by Actuate. For installations with both ReportCast Agent and ReportCast Server on the same machine, if the web server's document root is wwwroot, the path to the default templates is:

```
wwwroot\actuate\<config>\standard
```

where <config> is the name of the configuration. For installations with separate ReportCast Agent and ReportCast Server installations, if the Actuate distribution directory is /usr/actuate, the path to the default templates is:

```
/usr/actuate/ReportCast/templates/<config>/standard
```

where <config> is the name of the configuration.

Search order for web templates

ReportCast searches for template files in a specific order:

- e.Reporting Server folder

- e.Reporting Server's _ _WebAgent directory
- The Custom directory for a particular configuration
- The Standard directory for the configuration

For more detailed information about the search order, see “Understanding ReportCast searches for templates,” later in this chapter.

About the ReportCast object model

To understand how ReportCast's directives, scripting language commands, and templates work together, it is useful to understand something about the objects with which ReportCast works. ReportCast objects represent the URL requests, web server, e.Reporting Server, and objects on the e.Reporting Server. ReportCast performs actions on e.Reporting Server objects or displays their properties.

ReportCast objects

ReportCast operates on these types of objects.

Object	Description
Web server	Web server running ReportCast.
ReportCast	ReportCast process.
Web request	Request that ReportCast is currently handling.
e.Reporting Server	Actuate e.Reporting Server on which the requested object is located.
Folder	Folder in a Report Encyclopedia. Folders can contain other folders as well as files.
Folder item	One of the contents of a folder.
Report definition	.ROX, .ROV, .ROI, .ROW file types.
Request folder	A folder containing active, scheduled, or completed requests.
Request	A request to run a report. Requests are located in Request folders.
Member	A user, role, or group.
Parameter group	A group containing parameters for a specific report request.
Parameter	A specific parameter used to create a request.
Schedule	The part of a request's schedule that describes the times at which the request is to run.

Object	Description
Printer property	The part of the print request that provides information about the printer that will print the requested report.
Channel	An object that contains a set of requests.
Error	An object that contains information about an error encountered by ReportCast.
TOC item	An object that consists of a displayable name and a component ID.
View session	An object that tracks the current page, navigation mode, format, and other viewing properties for the current report.

Working with the ReportCast object model

The following table shows the relationships among ReportCast objects. The “Provides a list of” column lists the objects with which the objects in the Object column are associated.

Object	Provides a list of...
Web server	Nothing
ReportCast	Nothing
Web request	Nothing
e.Reporting Server	Members, request folders, channels
Folder	Folder items, members (those granted privileges)
Folder item	Depends on the kind of item
Report definition	Parameters, parameter groups, parameter values
Request	Request status
Request folder	Requests
Member	Privileges within the context of a folder item or channel
Parameter	Nothing
Parameter group	Parameter
Parameter value	Nothing
Privilege	Nothing
Channel	Requests
Error	Nothing
View session	

Object	Provides a list of...
TOC item	
Search criterion	
Search result set	
Search result hit	
Search result field	

Understanding objects available to a template

When you create templates, it is convenient to know when particular objects are available.

These objects are always available:

- e.Reporting Server (if specified in the URL)
- ReportCast
- Web request
- Web server

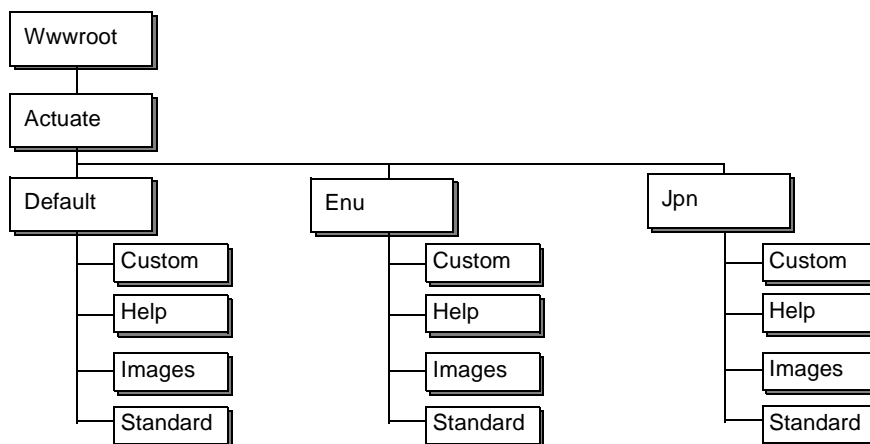
These rules determine when other objects are accessible. If the object does not meet one of the following requirements, it is not accessible:

- If an object is on the web page of the web server, ReportCast, request, or e.Reporting Server, it is available for that page's template.
- List objects are available to list all objects related to a specific object, such as a list of privileges, e.Reporting Servers, and so on. Each iteration through the list loop makes the next object in the list available.
- If an object is accessible, you can make its variables and list statements accessible as well using the With command. The With command has no effect if the object is not already accessible. For more information about the With command, see "With command," in Chapter 5, "ReportCast scripting language reference."

About the web server template directory structure

When you install ReportCast, the installation process sets up a template directory structure. The structure depends upon the type of ReportCast installation.

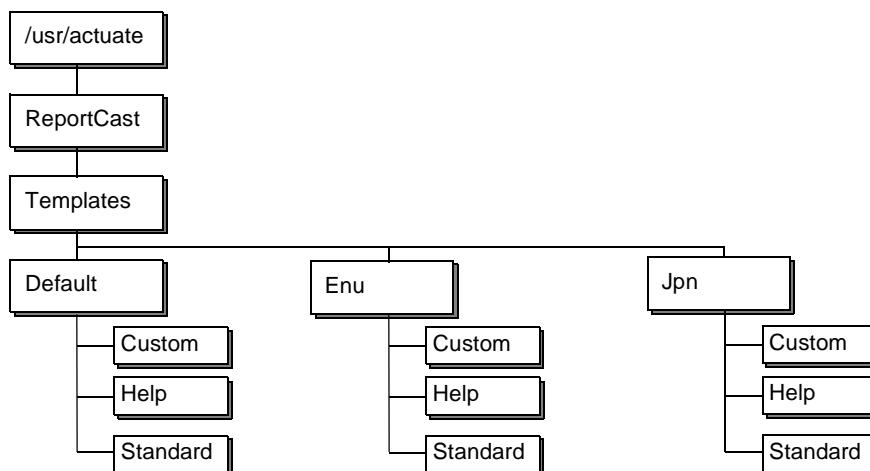
For ReportCast Server, ReportCast Agent, and installations with ReportCast Agent and ReportCast Server on the same machine, the structure is similar to the following diagram.



In this diagram, `wwwroot` represents the document root for your web server.

The directories `jpn` and `enu` are standard abbreviations for localized languages. The directory `default` is the default ReportCast configuration. If you do not specify a configuration, ReportCast uses the default directory.

For split ReportCast Server installations, the structure is similar to the following diagram.



In this diagram, `/usr/actuate` represents the root of your Actuate distribution directory structure.

ReportCast installations with ReportCast Agent and ReportCast Server on separate machines have the following additional considerations:

- Static content, such as JavaScript files and images, reside on the web server with the ReportCast Agent.
- The .ACHTML template files reside in the Actuate distribution directory structure with the ReportCast Server.
- To create custom ReportCast configurations with a split ReportCast installation, you must customize JavaScript files and images on the ReportCast Agent machine, and customize templates on the ReportCast Server machine.

The installation process automatically creates a directory structure for each language that ReportCast supports. The default template files go in the standard directories. Copy the templates you want to customize to the appropriate custom directory, and make changes to them there. This process ensures that you always have the original default templates. The help directories contain online help files for ReportCast. The images directories contain the images and icons that display on the reporting web pages, such as the folder, schedule, and search icons.

Creating and customizing ReportCast template files

Create and customize ReportCast template files in the same way that you work with any web page: use an editing tool or an HTML authoring tool. Most authoring tools do not support the ReportCast scripting language. You can use authoring tools to design your ReportCast pages, then add the scripting commands later. If your authoring tool supports adding HTML comments to your web pages, use that feature to add the scripting commands.

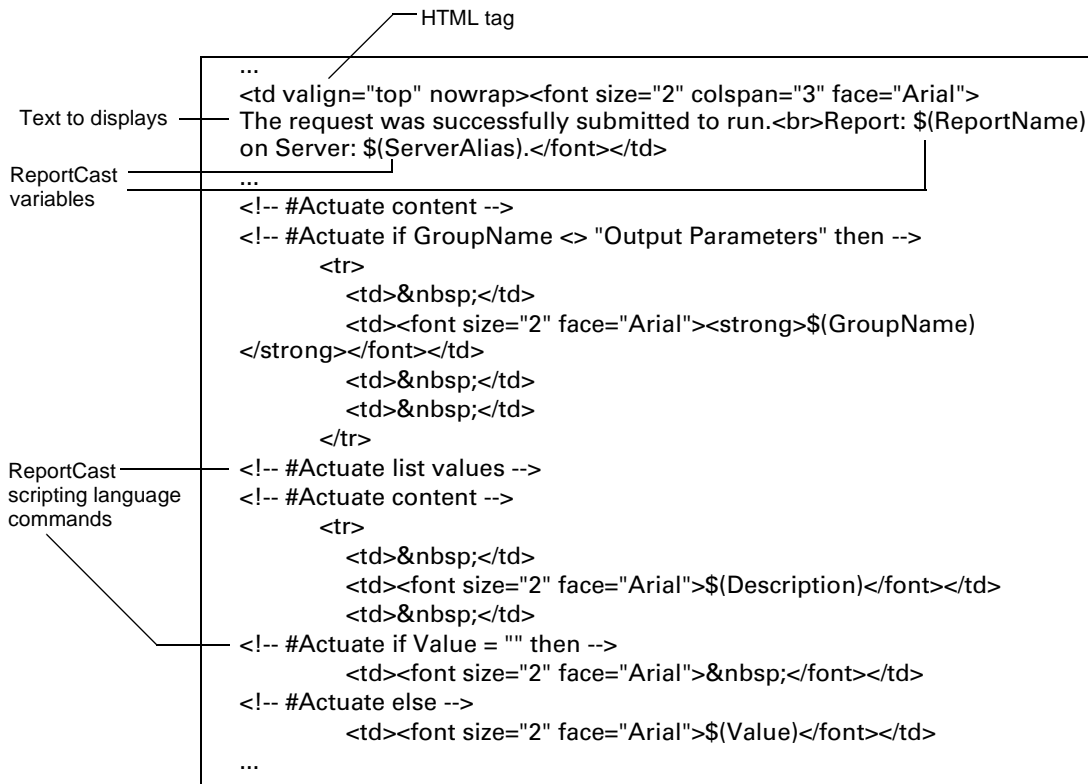
Template file elements

The default HTML template files contain these elements:

- HTML tags that determine text format
- Text
- Variables, which ReportCast substitutes with data according to the directive you issue. These variables appear in the following format:
`$ (VARIABLENAME)`
- ReportCast scripting language directives. These directives appear in the following format:
`<!-- #Actuate <directive> [options] -->`
- Javascript

The following is a fragment of the HTML document source for the CONTENTCONFIRM.ACHTML file, a template file that generates the content

portion of the request confirmation web page. The request confirmation page displays after a user submits a request to run a report.



Modifying HTML template files

You can easily modify template files written in HTML, to change the content and format of the HTML pages generated by ReportCast. For example, you can change the background pattern or color, or replace the Actuate bitmap in the HTML page. You can also add or delete variables in the template files to change the data.

Before editing any of these template files, you should:

- Be well-versed in HTML.
- Understand how to use ReportCast variables in templates.
- Be familiar with the ReportCast scripting language.
- Be familiar with Javascript.

For more detailed information about ReportCast scripting language variables, see “ReportCast scripting language variables,” in Chapter 5, “ReportCast scripting language reference.” For more detailed information about descriptions of the ReportCast scripting language, see Chapter 5, “ReportCast scripting language reference.”

Locating ReportCast template files

ReportCast expects:

- To find the templates in specific locations
- Templates to have specific names

If you put your templates in different locations, or give them different names, ReportCast will not find them. If ReportCast does not find the appropriate template, that template cannot be used to generate web pages. If you change the name of an included subtemplate file, you must change all references to that subtemplate file in all the ReportCast templates that refer to it. For more information about naming ReportCast template files, see “Naming template files,” later in this chapter.

Understanding ReportCast searches for templates

When it receives a directive, ReportCast first determines whether or not to look for custom templates. It then searches for custom templates in a specific order:

- 1 Determine the search type. The request may be to display folder contents or to run a report:
 - If this is a request to display folder contents, look in the e.Reporting Server folder for an INDEX.ACHTML file. If there is an INDEX.ACHTML file, use it to generate the folder contents web page.
 - If this is a request to run a report, look in the e.Reporting Server folder for an HTML file that depends on the report’s ROX. If there is one, assume that it is a custom requester form for the report and display it as the requester web page. If there is more than one HTML file that depends on a particular ROX, display a requester menu form. The user chooses the form.
- 2 Search for template files in the e.Reporting Server’s `__WebAgent` folder. Templates in the `__WebAgent` folder generate customized web pages for this e.Reporting Server.
- 3 For installations with ReportCast Agent and ReportCast Server on the same machine, search for template files in the web server’s Custom directory for the configuration currently in use. For installations with ReportCast Agent and ReportCast Server on separate machines, search for

template files in the ReportCast Server's Templates\<config>\Custom directory in the Actuate distribution directory, where <config> is the configuration currently in use.

Templates in the Custom directory generate web pages for this particular configuration. For example, if the configuration in use is NT ReportCast's French configuration, ReportCast uses the custom templates in wwwroot\Actuate\Frn\Custom to generate French-language web pages.

- 4 For installations with ReportCast Agent and ReportCast Server on the same machine, search for template files in the web server's Standard directory. For installations with ReportCast Agent and ReportCast Server on separate machines, search for template files in the ReportCast Server's Templates\<config>\Standard directory in the Actuate distribution directory, where <config> is the configuration currently in use.

If no other template files exist, ReportCast uses the default templates in the Standard directory.

You can develop custom templates for a particular e.Reporting Server or web server to create a distinctive look for a departmental e.Reporting Server or web server. Custom templates ensure that the department's web pages have a consistent design across your company's intranet.

For installations with ReportCast Agent and ReportCast Server on the same machine, the default template files are located in the Actuate directory within the web server's document root directory. For example, if your web server's document root is wwwroot, the default template files are in wwwroot\Actuate\Default\Standard.

For installations with ReportCast Agent and ReportCast Server on different machines, the default template files are located in the ReportCast server's ReportCast\Templates\Default\Standard directory of your Actuate distribution directory structure.

The following table lists the expected locations for ReportCast template files. Docroot refers to the document root, whether on the web server, or in the ReportCast Server directory structure.

Location	Description
Docroot\[config]Standard	Location of the default templates for configuration [config] of this web server. The default directory contains the default Actuate ReportCast configuration.
Docroot\[config]\Custom	Location of custom templates that you create for configuration [config] of this web server.

Location	Description
_ _WebAgent	Location of custom templates that you create for this e.Reporting Server. Use the e.Reporting Server Navigator to create this folder on your e.Reporting Server.
INDEX.ACHTML	The custom template you create specifically for a particular folder on your e.Reporting Server. Use the e.Reporting Server Navigator to place this template file in the appropriate folder on your e.Reporting Server.

A template's location—in an e.Reporting Server's WebAgent folder, in a custom directory, and so on—determines its scope. For more information about placement of templates for customizing part or all of the ReportCast interface, see "About template placement and scope," earlier in this chapter.

Creating custom requester forms for a specific report

You can create custom requester forms for a particular report. If an .ROX or an .ROV has one or more HTML files that depend on it, ReportCast treats those files as custom requester forms. When a request to run one of those reports occurs, ReportCast displays the custom requester form rather than the standard form. If there is more than one custom requester form, ReportCast displays a menu and allows the user to choose the form to use.

How to create a custom requester form

- 1 Use the editing tool of your choice to create a custom HTML requester form, perhaps using the standard requester, REQUEST.ACHTML, as a model.
- 2 Save the requester form with the extension .HTML.
- 3 In the Administrator Desktop, navigate to the e.Reporting Server folder containing the report. That is where you attach the custom HTML requester form.

How to attach a custom requester form to a report

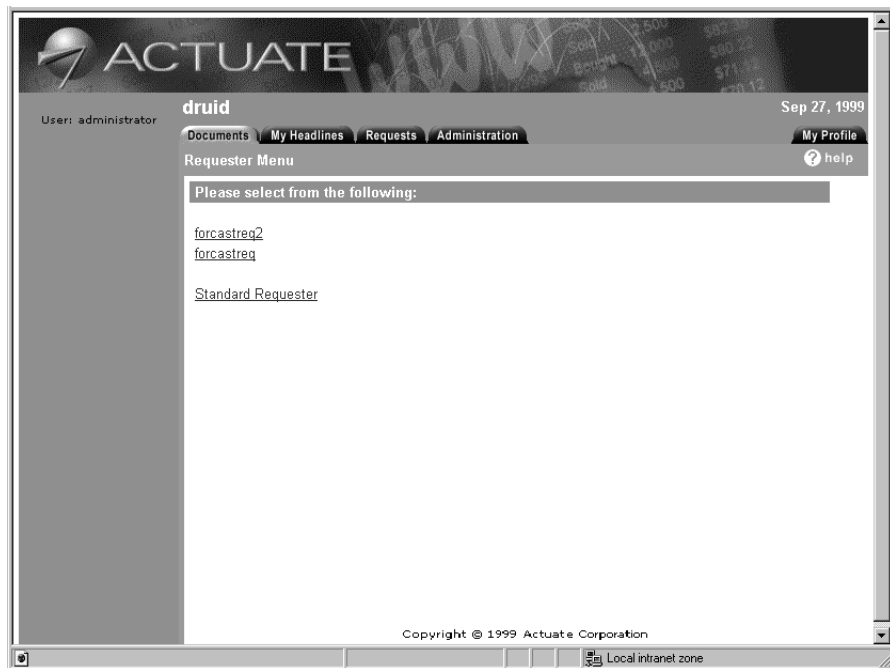
- 1 Select the e.Reporting Server folder that contains the report.
- 2 Right-click to access the Context menu.
- 3 Choose New Report Item from the Context menu.
The Save To Volume dialog box displays.
- 4 Locate the custom requester. Save it to the current folder.

- 5 Select the custom requester.
- 6 Right-click to access the Context menu.
- 7 Choose Properties from the Context menu.
- 8 Choose the Dependency tab.

The Dependency window displays.

- 9 Choose Add to add the .ROX as a dependency for the custom requester.
- 10 Choose OK.

If there is more than one custom HTML requester form for a particular .ROX, ReportCast displays a form similar to the following when it receives a request to run a report.



How to select a requester form

The user selects the appropriate requester form link to use that form.

Reloading template files

Changes to the template files are not reflected in the generated HTML pages immediately. You must reload the templates from the ReportCast

Administration page to see the changes. Only users with administrative privileges can access the ReportCast Administration page.

How to reload templates

- 1 To access the ReportCast Administration page, either choose the Administration tab, or type the URL in your browser. The URL is similar to:
`http://<webserver>/acweb/_ _admin.`
where <webserver> is the name of the machine that runs your web server. If you choose the Administration tab, the ReportCast Channel Administration page appears. Click the ReportCast administration link on the left side of the page. The ReportCast administration page appears.
- 2 Click the Reload Templates link.
The Confirmation page displays if the templates reload successfully.
- 3 Verify your changes by displaying the appropriate ReportCast page.

Naming template files

The following table lists the names of the ReportCast template files. All templates have the file extension .ACHTML. All templates are in the Standard directory of the web server's Actuate directory. You must follow this naming convention for any custom template files you create or ReportCast will not find them.

Template	Description
Active	Displays active requests.
Activedetail	Lists the properties for an active request.
Admin	Displays the ReportCast Administration page.
Cancel	Displays confirmation that a request has been cancelled.
Channeladmin	Displays the ReportCast Channel Administration page.
Channelcontents	Displays the contents of a particular channel.
Channeldetail	Lists the properties for a particular channel.
Channelitemdetail	Lists the properties for a particular channel item.
Channelsubscription	Displays the ReportCast channels Subscription page.
Confirm	Displays confirmation that a request to run a report was submitted to e.Reporting Server.

Template	Description
Error	Displays information about an error condition.
Filedetail	Lists the properties for a particular report item.
Fileversions	Lists the available versions of the current file.
Folderlist	One-page view of a Report Encyclopedia folder.
Index	Custom folder page created by the administrator, and stored directly on the e.Reporting Server.
Personalchcontents	Displays the contents of a particular user's Personal Channel folder.
Request	Requester page generated by ReportCast.
Requestdetail	Displays the details for a particular request.
Requestmenu	Lists the available requester forms. This page displays only if there are custom requester forms available.
Requestlist	Lists the Active, Scheduled, or Completed requests.
Searchresults	Displays the results of searches on DHTML reports.
Scheduled	Lists all scheduled requests on the current e.Reporting Server.
Scheduleddetail	Lists the properties of a scheduled request.
Status	Displays the status of an active or completed request.
Success	Displays confirmation that a Drop or Rename operation completed successfully.
Viewframeset	Displays the frameset for viewing the DHTML report.
Viewnav	Composes the toolbar used for viewing DHTML reports.
Viewtoc	Displays the DHTML report's table of contents.

The following table maps ReportCast directives to the template(s) they generate or affect.

Directive	Template
--_active	Requestlist
--_completed	Requestlist

Directive	Template
__scheduled	Requestlist
Cancel	Cancel
__channels?admin	Channeladmin
__channels?retrieveForm	Channelsubscription
__channels?submitForm	Confirm
__channels?retrieveCreation	Channeldetail
__channels?retrieveDetails	Channeldetail
__channels/channel?drop	Success
DeleteStatus	Success
Detail	Filedetail
Request	Requestermenu
Sort and Direction	Folderlist
StdRequest	Request
Submit	Confirm
Drop	Cancel
FlushConnections	Admin
ViewFrameset	Viewframeset
GetDynamicData	Viewframeset
ViewPage	Viewframeset
GetReportData	
GetStyleSheet	Viewframeset
ViewTOC	Viewtoc
ViewNavigation	Viewnav
Versions	Fileversions
requestSearch	Searchrequest
submitSearch	Searchresults

Placing custom templates in a Report Encyclopedia

When you want to create a distinctive design for a particular department's e.Reporting Server web pages, use the Administrator Desktop's Navigator to place custom template files directly in the e.Reporting Server Encyclopedia. You must have administrator privileges on both the web server and the e.Reporting Server to perform this procedure.

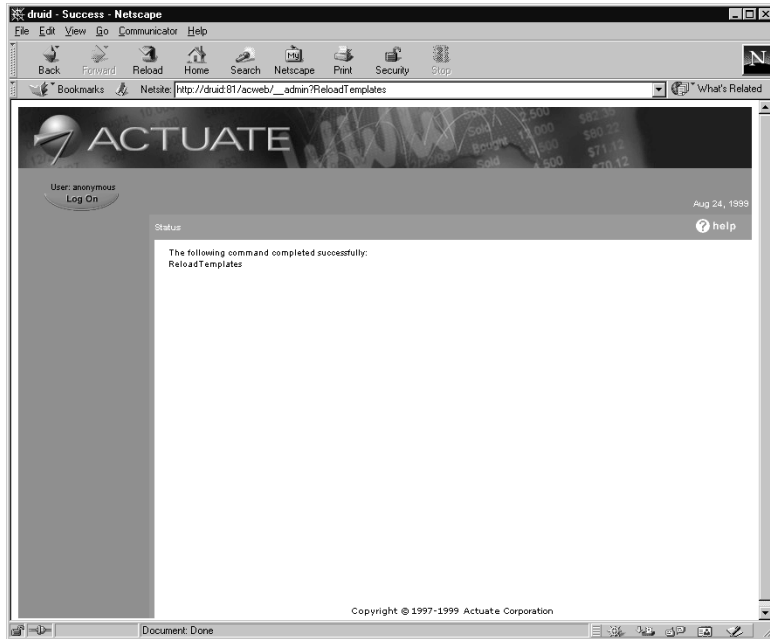
How to place custom templates in a Report Encyclopedia

- 1 Create the __WebAgent folder on your Report server:
 - 1 Start Actuate Administrator Desktop.
 - 2 Choose File→Navigator.
The Report Encyclopedia Login dialog box appears.
 - 3 Enter your login information.
 - 4 Choose OK.
 - 5 Choose the e.Reporting Server destination for the custom ReportCast templates.
 - 6 Choose File→New Encyclopedia Item→Folder.
The Create a new folder dialog box appears.
 - 7 Type __WebAgent in the Enter folder name box. Choose OK.
The __WebAgent folder appears in the Navigator windows.
- 2 Create custom templates for the Report Encyclopedia:
 - 1 Copy the default templates from the web server configuration's Standard directory to a work directory.
 - 2 Modify the templates.
 - 3 In the Navigator, open the __WebAgent folder.
 - 4 Choose File→New Encyclopedia Item→Report Item.
The Save to Volume dialog box appears.
 - 5 In the Files of Type box, choose All Files (*.*) .
 - 6 Browse to the location of your custom templates.
 - 7 Select a template to place in the __WebAgent folder.
 - 8 Choose Save to Volume.
The template appears in the Navigator window's __WebAgent folder contents listing.

Repeat steps 4 and 5 for each template you want to place in the __WebAgent folder.

To view the HTML source of a page you see in the Netscape browser, choose View→Document Source. The source document identifies the template file ReportCast used to generate the page.

The following illustrations show the source for the SUCCESS.ACHTML template file and the web page it generates. The complete Success web page is similar to the following page.



The following sections show the template source code that generates specific parts of the Success web page.

Web page title, background, banner, page header, username, date, and sidebar graphics



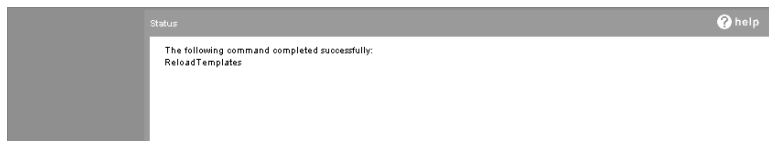
```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML/EN">
<html>
<!-- Actuate Template File: $(TemplateName) -->
<!-- #Actuate Include "header.achtml" -->
<head>
<!-- #Actuate Include "browsertest.achtml" -->
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<Meta Name="Generator" Content="Actuate Web Agent 3.0">
<title>$(Host) - Success</title>
<!-- #Actuate Include "headerscript.achtml" -->
</head>
<!-- #Actuate Include "bodybegin.achtml" -->
<!-- #Actuate Include "banner.achtml" -->
```

```

<!-- PageHeader Begin -->
<!-- #Actuate Set CurrentTab = tabNone -->
<!-- #Actuate Set ServerString = "" -->
<!-- #Actuate Include "pageheader.achtml" -->
<!-- PageHeader End -->
<!-- SideBar Begin -->
<!-- #Actuate Include "sidebarbegin.achtml" -->
<!-- #Actuate Include "sidebarsuccess.achtml" -->
<!-- #Actuate Include "sidebarend.achtml" -->
<!-- SideBar End -->

```

Display Status, result of command execution, command executed



```

<!-- #Actuate Include "contentheader.achtml" -->
<!-- PageTitle Begin -->
<!-- #Actuate Set PageTitle1 = "Status" -->
<!-- #Actuate Set PageTitle2 = "" -->
<!-- #Actuate Set PageTitle3 = "" -->
<!-- #Actuate Set PageTitle4 = "" -->
<!-- #Actuate Set PageTitle5 = "" -->
<!-- #Actuate Include "pagetitle.achtml" -->
<!-- PageTitle End -->
<!-- Content Begin -->
<!-- #Actuate Include "contentbegin.achtml" -->
<!-- #Actuate Include "contentsuccess.achtml" -->
<!-- #Actuate Include "contentend.achtml" -->
<!-- Content End -->

```

Display copyright information, page footer, close page elements



```

<!-- #Actuate Include "contentfooter.achtml" -->
<!-- #Actuate Include "pagefooter.achtml" -->
<!-- #Actuate Include "bodyend.achtml" -->
<!-- #Actuate Include "metaend.achtml" -->
</html>

```

For more information about ReportCast scripting language variables, see “ReportCast scripting language variables,” in Chapter 5, “ReportCast scripting language reference.”

Creating custom templates

This section provides examples of customizing the HTML pages generated by ReportCast.

How to customize a particular template

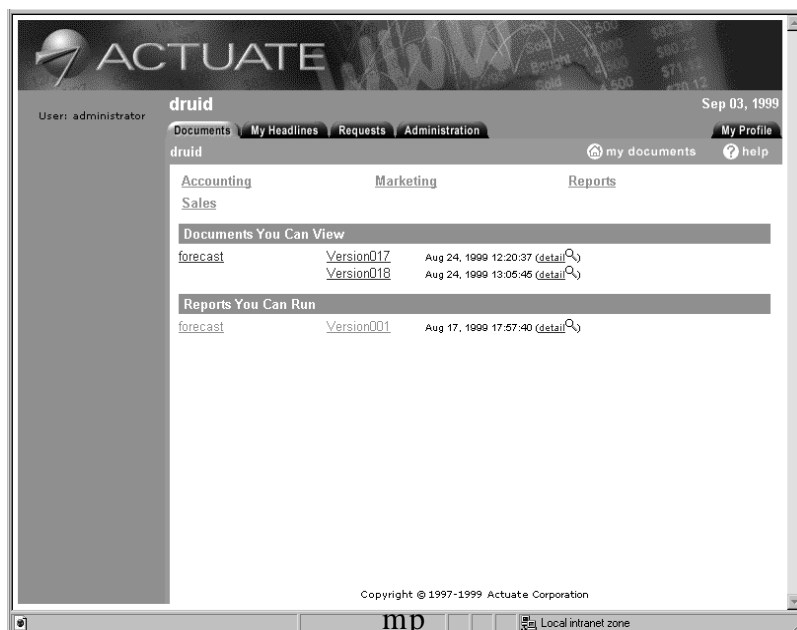
- 1 Display the standard web page for the template you want to customize.
- 2 In your web browser, go to a Report Encyclopedia folder containing report executables.
- 3 Click a report executable.
The Requester web page displays in your browser.
- 4 Choose File→Save As.
Save the web page in your configuration’s custom directory.
- 5 Edit the file in the Custom directory.

The standard templates include other template files and use variable substitution to display appropriate banners and icons on the resulting web page. The advantage to this method is that various elements are easily reused by several templates. The disadvantage is that elements such as nested included files can be confusing.

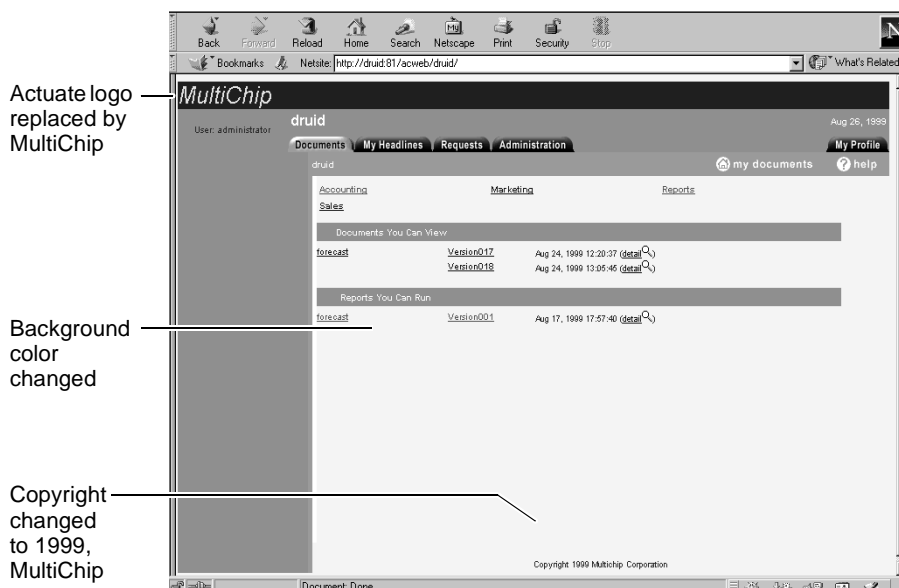
You can display the page you wish to customize in the web browser, then save it. The web page source is much more straightforward, substitutions have been made and the contents of included files are specifically included. You can edit and customize the complete web page.

Example This example shows how you can modify template files to change the format and data that ReportCast generates when it receives an `http://.../<folder>` directive. The following illustration shows the HTML pages generated from the default and custom template files.

Default HTML page generated from a http://.../ <folder> directive



Customized HTML page



Modifying templates

To make the changes shown in this example, you modify these files:

- FOLDERLIST.ACHTML, the template that generates Folder listing pages.
- CONTENTFOLDERLIST.ACHTML, included by FOLDERLIST.ACHTML to generate the content for the Folder listing pages.
- HEADER.ACHTML, included by FOLDERLIST.ACHTML to define reporting web page colors, element names, commonly used URL's, and locations of images.
- HEADERSCRIPT.ACHTML, a Javascript file included by FOLDERLIST.ACHTML to preload images for the reporting web page.
- BANNER.ACHTML, included by FOLDERLIST.ACHTML to generate the company logo.
- COPYRIGHT.ACHTML, included by FOLDERLIST.ACHTML to generate the copyright.

Before you get started, create a directory called Custom in your configuration directory. For example, if you are using the U.S. English configuration, create the custom directory in your web server's `actuate\enu` directory.

You must reload the templates to activate your changes. Reload templates from the ReportCast administration page, or use the `ReloadTemplates` directive.

How to replace one logo with another

Because the following is such a common requirement, the following steps take two sample logos and describe how to replace the Actuate logo with the Multichip logo:

- 1 Copy BANNER.ACHTML from the Standard directory to the custom directory.
- 2 Edit the file. The default contents are:

```
<tr>
  <td valign="top" colspan=1 bgcolor="$(colorLogoBack)" nowrap>
    </td>
    <td width="100%" valign="top" colspan=6 bgcolor="$(colorLogoBack)"
      nowrap>
    </td>
</tr>
```

The images `imgLogo1` and `imgLogo2` generate the default Actuate logo and banner for all the reporting web pages. The `HEADER.ACHTML` template defines `imgLogo1` and `imgLogo2`.

- 3 Replace the HTML code:

```

```

as follows:

```
<font color="yellow" size="6" face="Courier"><strong><em>
MultiChip Corporation</em></strong></em></font>
```

- 4 Replace the HTML code:

```

```

as follows:

```
&nbsp;
```

- 5 Save BANNER.ACHTML.

- 6 View the changes:

- 1 Reload the templates.
- 2 Return to the ReportCast home page. The home page is now similar to the following image.



How to change the border color

- 1 Copy the FOLDERLIST.ACHTML, PAGEHEADERX.ACHTML, and HEADER.ACHTML templates to the custom directory.
- 2 Edit PAGEHEADERX.ACHTML:
 - 1 Search for the string BGCOLOR. Search results include lines such as this one:

```
<TD HEIGHT="10" COLSPAN="6" VALIGN="bottom"
BGCOLOR="$(colorLightBlue)">
```

- 2 Replace the color (\$colorLightBlue) with turquoise. For example:

```
<TD HEIGHT="10" COLSPAN="6" VALIGN="bottom" BGCOLOR="turquoise">
```

- 3 Save PAGEHEADEREX.ACHTML.

- 3 Edit FOLDERLIST.ACHTML:

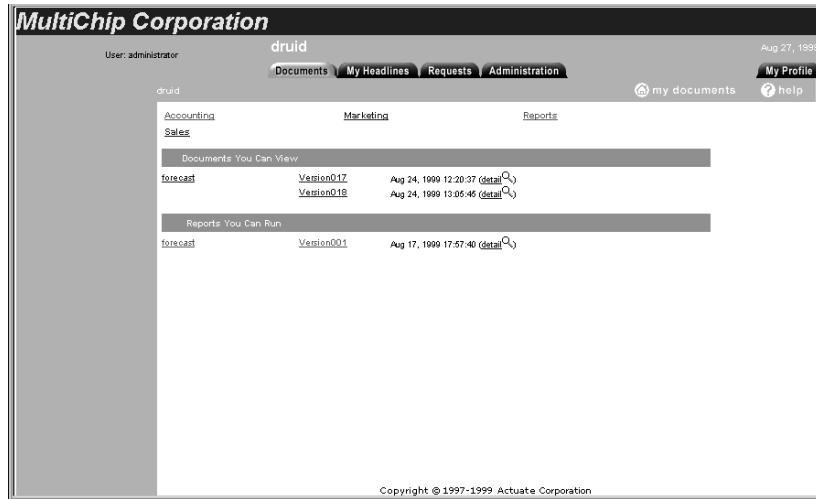
Search for the string BGCOLOR. Search results include lines such as this one:

```
<TD WIDTH="149" BGCOLOR="$(colorLightBlue)" VALIGN="TOP"
NOWRAP><IMG SRC="$(imgSpace)" HEIGHT="5" WIDTH="149"></TD>
```

Replace the color (\$colorLightBlue) with turquoise. For example:

```
<TD WIDTH="149" BGCOLOR="turquoise" VALIGN="TOP" NOWRAP>
<IMG SRC="$ (imgSpace)" HEIGHT="5" WIDTH="149"></TD>
```

- 4 Save FOLDERLIST.ACHTML.
- 5 Reload the templates. The web page looks similar to the following graphic.



The steps in the previous procedure result in a turquoise border for the Folder listing page. You can also include custom background images or set custom background colors for your Folder listing page.

The HEADER.ACHTML file contains settings for all the templates' image variables.

How to change the copyright information

The standard templates use a copyright image to display copyright information on the reporting web pages. To change the copyright information, change the image file, IMG_COPYRIGHT.GIF, in the images directory.

Customizing the e.Analysis add-on option

The e.Analysis application is an add-on product that works with DHTML reports on the Web. You use e.Analysis to analyze data fields selected in a Smart Search. For more information, see *Using e.Analysis*.

If e.Analysis is installed, you can customize whether the Analyze Results button appears or whether Analysis appears on the Download as a menu item. The SEARCHRESULTS.ACHTML template controls whether the Analyze

Results button appears or whether Analysis appears on the Download as a menu item.

How to change where the user launches e.Analysis

To change where the user launches e.Analysis, you modify the SEARCHRESULTS.ACHTML template. If your SEARCHRESULTS.ACHTML template is in the custom directory of your default Actuate web reporting configuration directory, that template is the one you must modify. If your SEARCHRESULTS.ACHTML template file in is not in the custom directory of your default Actuate web reporting configuration directory, you must modify the SEARCHRESULTS.ACHTML in the standard directory of your default Actuate web reporting configuration directory.

- 1 Using a text editor, open the SEARCHRESULTS.ACHTML template file.
- 2 Find the following two variables in the SEARCHRESULTS.ACHTML template file:
 - var showAnalysis
 - var showAnalysisButton
- 3 To change the visibility of e.Analysis, set the values of the variables based on the following table.

e.Analysis visibility	Statement in template file
To show Analyze Results button	var showAnalysis = true; var showAnalysisButton = true;
To show Analysis on drop-down menu	var showAnalysis = true; var showAnalysisButton = false;
To hide e.Analysis from user	var showAnalysis = false; var showAnalysisButton = true;

- 4 Reload the ReportCast templates:

http://<webserver>acweb<report server>__admin?ReloadTemplates

Where <webserver> is the name of your web server, and <report server> is the name of the e.Reporting Server. When users complete a Smart Search, the Smart Search page displays your configuration for the launch of e.Analysis.

How to enable launching e.Analysis

The e.Analysis install places the SEARCHRESULTS.ACHTML template and the SEARCHRESULTS.ACHTML.NEW template in the default configuration's Standard directory. If you have a modified SEARCHRESULTS.ACHTML template in the Custom directory, the Analyze Results button does not appear on the Search Results page and you cannot launch e.Analysis.

To add the e.Analysis button to your modified SEARCHRESULTS.ACHTML template:

- 1 Open the SEARCHRESULTS.ACHTML.NEW template located in the <wwwroot>\Actuate\default\standard directory.
This template contains the information you need to enable e.Analysis.
- 2 Open the SEARCHRESULTS.ACHTML template located in the <wwwroot>\Actuate\default\custom directory.
- 3 Add the code differences from the SEARCHRESULTS.ACHTML.NEW in the Standard directory to the SEARCHRESULTS.ACHTML template file in the Custom directory.
- 4 Find the variables showAnalysis and showAnalysisButton in the Custom directory's SEARCHRESULTS.ACHTML template and set the values as follows:

- var showAnalysis = true;
- var showAnalysisButton = true;

- 5 Reload the ReportCast templates:

`http://<webserver>acweb<report server>_ _admin?ReloadTemplates`

When users complete a Smart Search, the Smart Search page displays your configuration for the launch of e.Analysis.

Customizing the DHTML and SmartSearch displays

This section describes how to customize the DHTML and SmartSearch displays.

The following table lists the files that affect DHTML and SmartSearch display:

File	Description
Frameset	Determine the DHTML frameset look and feel
VIEWFRAMESET.ACHTML	
Toolbar	Determine the look and feel of the DHTML toolbar
VIEWNAV.ACHTML	
VIEWNAV.JS	

File	Description
Search	Determine the SmartSearch look and feel
SEARCHREQUEST.ACHTML	
SEARCHRESULTS.ACHTML	
SEARCH.JS	
Navbar (Table of Contents)	Determine the look and feel of the DHTML table of contents
VIEWTOC.ACHTML	
TOCTREE.JS	
Download as PDF	
Save as achtml	

ReportCast directives

This chapter contains the following topics:

- About ReportCast directives
- Types of ReportCast directives
- About Channel directives
- Examples of ReportCast directives

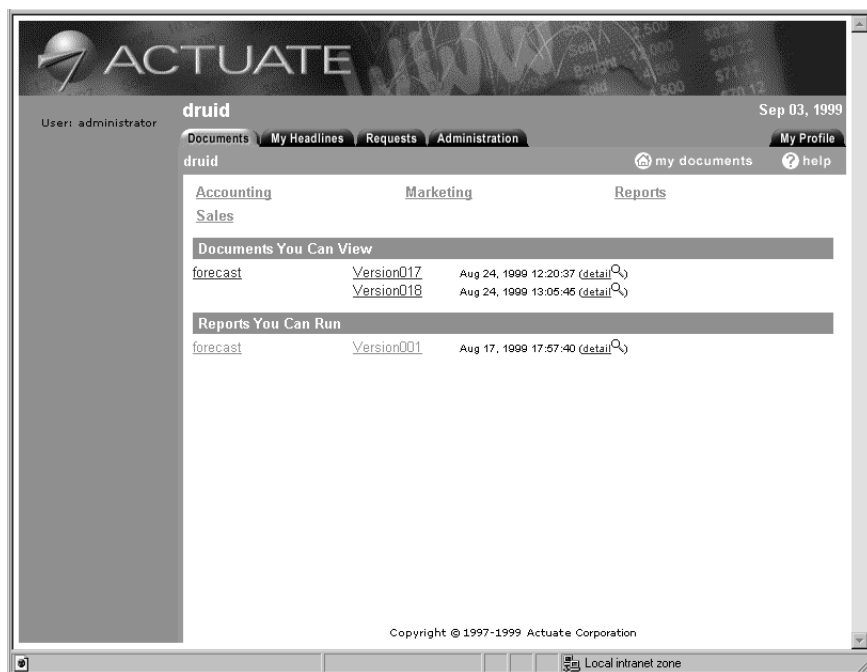
About ReportCast directives

ReportCast performs tasks based on directives from the web browser. Directives are URLs that contain the keyword `acweb` followed by the directive. When the web server encounters the `acweb` keyword, it passes the entire URL to ReportCast for processing.

Sending directives

Directives are often associated with buttons or links in the HTML pages generated by ReportCast. Choosing one of those buttons or links sends a directive. Another way to send a directive is to type the URLs in the Address or Location window of the web browser.

The following illustration shows a sample page generated by ReportCast. The page contains many links that, when clicked, send the appropriate URL directive to ReportCast.



You can see what directive is associated with a link by placing the cursor over the link.

How to view Netscape link information

The Netscape server displays complete URLs automatically.

How to view Microsoft Internet Explorer link information

To display complete URLs in Microsoft's Internet Explorer:

- 1 Select View→Options.
- 2 Choose Advanced.
- 3 Deselect the Show friendly URLs.

The URL appears in the status line at the bottom of the browser window.

Managing directives

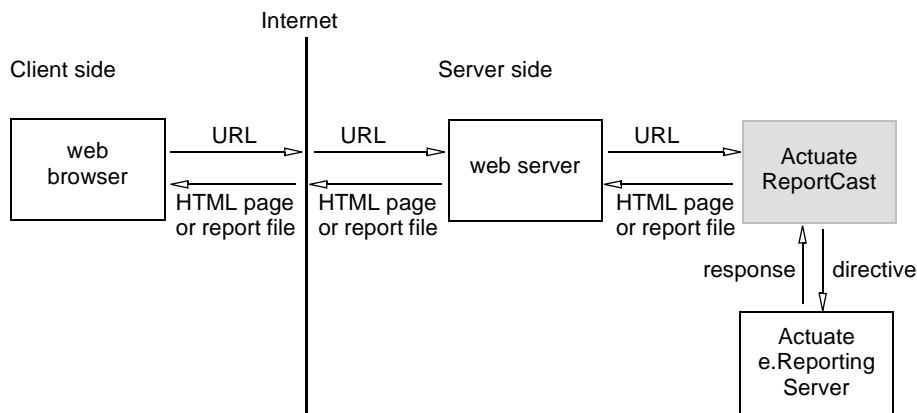
ReportCast uses HTML template files to generate HTML pages in response to directives sent from the web browser. Templates can be report-specific, folder-specific, e.Reporting Server-specific (apply to generating HTML pages on a particular e.Reporting Server), or web server-specific. ReportCast locates these template files by searching in a specific order:

- 1 The e.Reporting Server folder.
- 2 The e.Reporting Server's _ _WebAgent directory.
- 3 The web server's Custom directory for a particular configuration of the web server.
- 4 The web server's Standard directory.

For more detailed information about the search order, see "Understanding ReportCast searches for templates," in Chapter 3, "Working with ReportCast templates."

For detailed information about the HTML template files, see "About ReportCast templates," in Chapter 3, "Working with ReportCast templates."

The following illustration shows the flow of communication from the web browser to ReportCast and from ReportCast back to the web browser.



Formatting directives

ReportCast directives use one of the following formats:

```
{http | https}://<webserver>/acweb/ [<configuration>] <reportserver> | __local |
__default/ [<path>] [<resource>] [;<version>] [ ?<command> ]
```

or:

```
{http | https}://<webserver>/acweb/ __admin
```

In the UNIX environment, for CGI programs only, ReportCast directives use one of the following formats:

```
{http | https}://<webserver>/<cgi script location>/acweb/ [<configuration>]
<reportserver> | __local | __default/
 [<path>] [<resource>] [;<version>] [ ?<command> ]
```

or:

```
{http | https}://<webserver>/<cgi script location>/acweb/ __admin
```

The following section provides additional information about each of the parameters of the directives.

http | https

- Required.
- Specifies that the web browser and web server communicate using the HTTP communications protocol.

If your web server runs on the Secure Sockets Layer, URLs begin with https instead of http.

<webserver>

- Required.
- Name of the web server machine.

<cgi script location>

- Required to run ReportCast on UNIX using CGI programs.
- Not applicable to other platforms.

acweb

- Required.
- Tells the web server to pass the rest of this URL's fields to ReportCast.

__admin

- Optional.
- Directive that accesses the ReportCast Administration page.

<configuration>

- Optional.
- Name of the set of templates ReportCast should use to generate web pages for this request.

<reportserver> | __local | __default

- Required.
- Name of the e.Reporting Server you want to access.

If the e.Reporting Server, ReportCast Server, ReportCast Agent and web server are installed on the same machine, specify the literal `__local` instead. `__local` is the name of the web server machine that has connected to the ReportCast Server machine. This is the web server machine itself, not a proxy server. To resolve `__local` to a host name, DNS on the ReportCast Server machine must be able to find a host name for the address of the web server machine. Otherwise, `__local` is blank.

Specify the literal `__default` to access the e.Reporting Server on the machine specified in the `AC_REPORTCAST_DEFAULT_REPORT_SERVER` environment variable on UNIX, or registry key on NT.

ReportCast internally substitutes the e.Reporting Server's machine name for `__local` or `__default`. For example, if the user enters the following URL:

`http://webserver/acweb/__local`

ReportCast includes `__local` in all the links that it generates. The actual name of the machine displays in HTML. For example, ReportCast generates the following URL for a Report Encyclopedia folder called MyFolder:

`http://webserver/acweb/__local/MyFolder/.`

For information about setting the default e.Reporting Server, see “Setting ReportCast default e.Reporting Server,” in “Administering ReportCast.”

<path>

- Optional.
- Full path to the Report Encyclopedia folder or file to access.

<resource>

- Optional.
- A file or other resource in the Report Encyclopedia folder specified by <path>. If you do not specify this field, ReportCast accesses the folder specified by <path>.

;<version>

- Optional.
- Version number of the folder, file, or other resource to access.

?<command>

- Optional.
- Action to execute.

Types of ReportCast directives

The following table lists the types of ReportCast directives.

Type of directive	Description
Folder directive	Specifies how the contents of a Report Encyclopedia display.
File directive	Downloads report files from the e.Reporting Server.
Factory directive	Generates new reports from report executables (.ROX).
Status directive	Gets report request status.
Channel directive	Administers channels and retrieves channel contents.
User preferences directive	Displays the user preferences form and submits user preferences changes.

Type of directive	Description
Viewing directive	Determines how to view completed reports.
Search directive	Performs searches on DHTML reports.

ReportCast directives

The following table lists the ReportCast directives.

Directive	Description
__admin	Accesses the ReportCast administration page. Do not specify a report server in the URL when accessing the administration page.
__home	Accesses the current user's home folder in the Report Encyclopedia, as set in the Administrator Desktop.
?DeleteStatus	Deletes status of a report-generation request.
?Detail	Generates detailed information about a Report Encyclopedia object.
?Drop	Cancels the report-generation request. Deletes a Report Encyclopedia object.
?Files	Lists only the files the current folder contains. Does not list subfolders.
?Reports	Displays the reports in the specified folder, sorted by name.
<report URL>	Downloads the entire report document.
?Request	Generates requester forms.
?StdRequest	Generates the standard requester form.
?Submit	Submits a request. Starts the report generation process.
?Status	Displays status for the current report-generation request.
?Sort and Direction	Specifies how to display the contents of a Report Encyclopedia.
Versions	Displays a list of the available versions of a file when the file the user is viewing in DHTML becomes unavailable.

Directive	Description
?View	Displays a report, one page at a time, in a browser. For HTML reports, ?View displays the entire report in the browser.
<report URL> ?ViewDefault	Displays a report in a format determined by the user's preference and viewing privileges.
<report URL> ?ViewFrameset	Displays the report in DHTML.
--_scheduled	Displays the list of currently scheduled factory requests.
--_active	Displays the list of currently active factory requests.
--_completed	Displays the list of completed factory requests.
--_channels?admin	Displays the channels administration page.
--_channels? retrieveForm	Displays the subscription form for available channels.
--_channels? submitForm	Submits subscription information.
--_channels? retrieveCreation	Displays the form used to add new channels.
--_channels? submitCreation	Submits the new channel form.
--_channels/ channel?drop	Removes an existing channel.
--_channels/ channel?retrieveDetails	Displays a form used to modify channel attributes.
--_channels/ channel?submitDetails	Submits the channel details form.
--_requestUser Preferences	Requests the user preferences page.
--_submitUser Preferences	Submits the user preferences page as an HTML form.
<report URL> ?GetDynamic Data	Retrieves data for a specific component, such as a graph or image in a report.
<report URL> ?ViewPage	Retrieves a specific page of a report, either from the navigation bar or from a set of search results.
<report URL> ?GetReportData	Retrieves report data in a particular format.

Directive	Description
<report URL> ?GetStyleSheet	Retrieves the style sheet for the report in a particular format, such as CSS or XMLStyle.
<report URL> ?ViewTOC	Retrieves the report's table of contents from the view process.
<report URL> ?ViewNavigation	Displays the navigation bar for viewing a report.
<report URL>?Versions	Displays the list of available versions of a file, when the file the user is viewing in DHTML becomes unavailable.
<report URL> ?searchReport	Performs a search on a DHTML report according to criteria that either users specify from the DHTML search page, or that you specify in a URL that you embed in a report or web page.
<report URL> ?requestSearch	Displays the search request page.
<report URL> ?submitSearch	Submits the search request and returns the search results page.
<report URL> ?extractSearch Results	Submits the search and returns search results in a specified format, such as comma-separated values or tab-separated values.

Directives syntax

In the following sections, the discussion of directives groups the directives by the objects they affect. All directives begin with one of the following forms of syntax:

`http://<webserver>/acweb/<reportserver>/<folder>?`

`http://<webserver>/acweb/<reportserver>/<folder><file>?`

<webserver>

Names the machine that is running the web server.

acweb

Literal string indicating that the rest of this string must be passed to ReportCast for processing.

<reportserver>

Names the e.Reporting Server to access.

<folder>

Names the folder to access on the e.Reporting Server.

<file>

Names the report document to access on the e.Reporting Server.

?

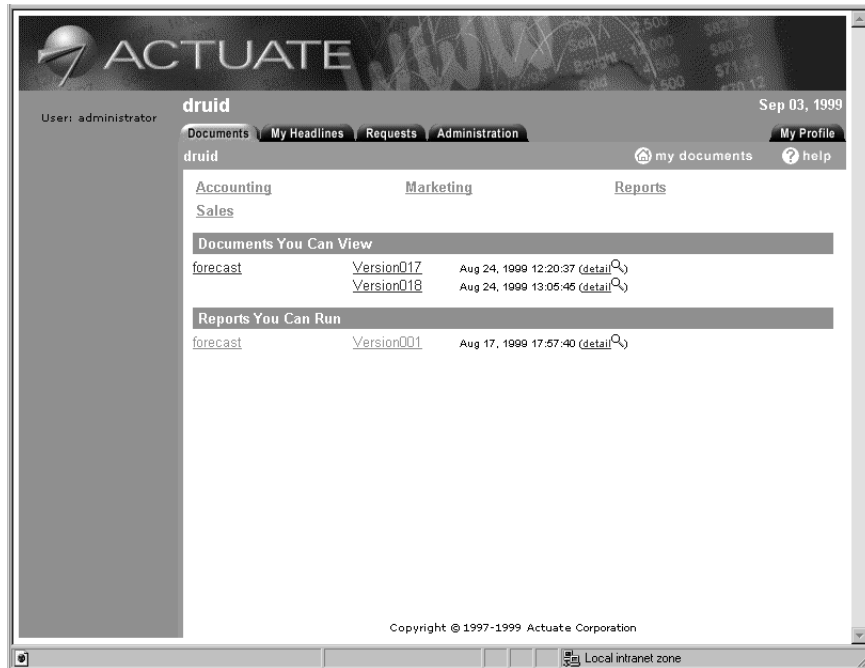
Indicates that the directive and the directive's parameters follow.

Folder directives

This section describes the folder directives that specify how to display the contents of a Report Encyclopedia.

About sort and direction directives

When you tell ReportCast to display the contents of a folder on the report server, it displays the folder's files sorted by name and version.



You can choose to display a folder's contents in different ways. For example, you can decide to view reports sorted by size, with the smallest reports listed first:

<http://mustique/acweb/mustique/Marketing?sort=size&direction=desc>

Using sort and direction directives

Use the sort directive to specify the sort order for a list of Report Encyclopedia folder contents. Use the direction directive to specify whether the sort order should be ascending or descending.

In the previous example, the URL displays Report Encyclopedia reports sorted by size, with the largest reports listed first.

Specify the sort and direction directives in any sequence. Remember to separate directives and their options with ampersands (&). The ampersand (&) is the standard command separator character for URLs.

Here is the syntax for the sort and direction directives:

```
http://<webserver>/acweb/<reportserver>/<folder> [?sort=<field>] [&direction  
= [desc | asc] ] [files]
```

For information about the general syntax of directives, see “Directives syntax,” earlier in this chapter. The following section provides additional information about each of the parameters of the sort and direction directives.

?sort=<field>

The sort directive, followed by the sort field. The following table lists the valid sort fields. These sort fields are associated with the Name, Type, Owner, Date, and Size links on the HTML page that displays the contents of a folder.

<field> value	Description
Name	Sorts the folder contents by name and version number
Type	Sorts the folder contents by file type
Owner	Sorts the folder contents by owner
Created	Sorts the folder contents by creation date
Date	Sorts the folder contents by modified date
Size	Sorts the folder contents by size

direction = [desc | asc]

The direction of the sort, descending (desc) or ascending (asc). Ascending is the default.

files

Lists only the files that this folder contains. Does not list subfolders.

Displaying only report files in a folder

The following table lists the directives you can send if you want to view only report files, not subfolders.

Directive	Displays ...
<code>http://<webserver>/acweb/<reportserver>/<folder>/?files</code>	Reports in the specified folder
<code>http://<webserver>/acweb/<reportserver>/<folder>/?files&sort=Name</code>	Reports in the specified folder, sorted by name
<code>http://<webserver>/acweb/<reportserver>/<folder>/?files&sort=Type</code>	Reports in the specified folder, sorted by file type
<code>http://<webserver>/acweb/<reportserver>/<folder>/?files&sort=Owner</code>	Reports in the specified folder, sorted by creator
<code>http://<webserver>/acweb/<reportserver>/<folder>/?files&sort=Date</code>	Reports in the specified folder, sorted by modified date
<code>http://<webserver>/acweb/<reportserver>/<folder>/?files&sort=Created</code>	Reports in the specified folder, sorted by creation date
<code>http://<webserver>/acweb/<reportserver>/<folder>/?files&sort=Size</code>	Reports in the specified folder, sorted by size
<code>http://<webserver>acweb/<reportserver>/<folder>?files&sort=Date&direction=desc</code>	Reports in the specified folder, sorted by creation date in descending order

Viewing DHTML reports

If Dynamic HTML (DHTML) is your report viewing preference, ReportCast displays reports in your web browser in DHTML format. This section describes the DHTML report viewing directives.

Displaying DHTML reports

The ViewFrameset directive generates the toolbar for report navigation and displays the initial frame set for viewing a report in DHTML format. Use the following syntax to display reports in DHTML format:

```
http://<webserver>/acweb/<reportserver>/<folder>/  
<file>?ViewFrameset&page=<page>
```

For information about the general syntax of directives, see “Directives syntax,” earlier in this chapter.

Use the page keyword to specify the number of the initial page to display. The default initial page number is the first page of the report.

You can bookmark DHTML report pages.

Users generally view DHTML report pages using the navigation bar. The navigation bar includes buttons for next, previous, and first and last pages.



Displaying available versions of DHTML reports

The Versions directive displays a list of the available versions of a report. Use the following syntax to display the list of available versions:

`http://<webserver>/acweb/<reportserver>/<folder>/<file>?Versions`

For information about the general syntax of directives, see “Directives syntax,” earlier in this chapter.

Available reports may expire after specific lengths of time. It is possible that a particular DHTML report could expire and become unavailable as you view it. When a report becomes unavailable, ReportCast displays a list of the available versions of the report. Click on the version you want to view. ReportCast displays that version of the report in DHTML format.

Getting dynamic content of DHTML report components

The GetDynamicData directive retrieves data for a component that the view process generates dynamically. The component is usually a graph or image in a report. Use the following syntax to retrieve a dynamically-generated component:

`http://<webserver>/acweb/<reportserver>/<folder>/
<file>?GetDynamicData&ComponentID=<ID>&scalingFactor=<scale>`

For information about the general syntax of directives, see “Directives syntax,” earlier in this chapter.

Use the componentID keyword to specify the ID of the dynamically-generated component.

Use the scalingFactor keyword to specify the size of the component, or the zooming percentage. A scaling factor of 100 indicates the original size of the component. A scaling factor of 50 displays the component at 50% of its original size. 100 is the default scaling factor.

Viewing a page from the navigation bar

The ViewPage directive allows printing or displaying a specific page of a DHTML report. Use the following syntax to prepare print or display output of a DHTML report page from the navigation bar:

```
http://<webserver>/acweb/<reportserver>/<folder>/  
    <file>?ViewPage&format=<format>&page=<page>&mode=<mode>  
    &scalingFactor=<scale>&operation=<operation>
```

For information about the general syntax of directives, see “Directives syntax,” earlier in this chapter.

Users generally display a specific page by typing the page number in the Page field of the navigation bar.

Users generally prepare print output of a specific page or range of pages by:

- 1 Clicking the Download/Print link in the navigation bar
- 2 Specifying the page or pages they wish to print in the Custom Report Saving Option form
- 3 Clicking the Apply button, then saving the print output to their local disk

The print output is PDF format. Preparing print output is the default. If your site requires that viewing is the default choice when using the Download/Print link, you must change the SAVEAS.ACHTML template. Change the following code:

```
operation=print  
to  
operation=view
```

Use the following syntax to display a page from a set of search results or from the report’s table of contents:

```
http://<webserver>/acweb/<reportserver>/<folder>/  
    <file>?ViewPage&format=<format>&componentID=<ID>  
    &scalingFactor=<scale>
```

Users generally display a page from a set of search results by clicking a link in the search results list.

The format keyword specifies the format in which to display the report. Format is any format that the view process supports, such as XML structure or page-view formats, or DHTML. The valid values for format are:

- DHTML—a compressed DHTML format that uses Cascading Style Sheets (CSS). This is the default format.
- DHTMLLong—an uncompressed DHTML format. Use this format if your browsers do not support CSS.

The mode keyword indicates the page to display, relative to the current page number. Values for mode are:

- First—displays the first page of the report
- Last—displays the last page of the report
- Previous—displays the page that precedes the current page
- Next—displays the page that follows the current page
- Specific—displays the page specified by the page keyword

The scalingFactor keyword functions as it does for the GetDynamicData directive, described in “Getting dynamic content of DHTML report components,” earlier in this chapter.

The operation keyword indicates whether to prepare PDF output for printing or to view PDF format report pages. The default is to prepare PDF format for printing reporting pages. The user then saves the prepared print output to disk and prints the saved output. If your site requires viewing PDF report pages as the default, you must change the SAVEAS.ACHTML template. Values for operation are:

- Print—Prepare PDF format print output and save the specified report page or pages to disk.
- View—Save the specified report page or pages as PDF and view the PDF output.

Getting report data

The GetReportData directive retrieves data for a report in a specified format. Use the following syntax to retrieve report data:

```
http://<webserver>/acweb/<reportserver>/<folder>/  
  <file>?GetReportData&format=<format>[&componentID=<ID>]&scalingFac  
  tor=<scale>&operation=<operation>.
```

For information about the general syntax of directives, see “Directives syntax,” earlier in this chapter.

The scalingFactor keyword functions as it does for the GetDynamicData directive, described in “Getting dynamic content of DHTML report components,” earlier in this chapter.

The operation keyword functions as it does for the ViewPage directive, described in “Viewing a page from the navigation bar,” earlier in this chapter.

If you do not specify a componentID, Actuate retrieves the entire report.

Getting the report style sheet

The GetStyleSheet directive retrieves the style sheet for the report in the specified format. Use the following syntax to retrieve the style sheet:

```
http://<webserver>/acweb/<reportserver>/<folder>/  
<file>?GetStyleSheet&format=<format>
```

For information about the general syntax of directives, see “Directives syntax,” earlier in this chapter.

The format keyword specifies the format in which to retrieve the style sheet. Valid values are CSS, for cascading style sheets for DHTML output, and XMLStyle, for style information specified in XML.

Viewing the table of contents

The ViewTOC directive displays the DHTML report’s table of contents. Use the following syntax to display the table of contents:

```
http://<webserver>/acweb/<reportserver>/<folder>/  
<file>?ViewTOC&format=<format>&componentID=<ID>
```

For information about the general syntax of directives, see “Directives syntax,” earlier in this chapter.

The format keyword specifies the format of the table of contents. XMLDisplay is currently the only supported format.

The componentID keyword specifies the component to use as the root of the table of contents. 0 is the default component ID.

Users generally display a table of contents for a DHTML report by clicking on the TOC icon in the navigation bar.

Viewing the navigation bar

The ViewNavigation directive displays the navigation bar, used for viewing DHTML reports. Use the following syntax to display the navigation bar:

```
http://<webserver>/acweb/<reportserver>/<folder>/  
<file>?ViewNavigation&page=<page>
```

For information about the general syntax of directives, see “Directives syntax,” earlier in this chapter.

The page keyword specifies the DHTML report page to display.

The navigation bar displays when a user clicks on a DHTML report document in a report encyclopedia folder.

Viewing reports according to user preferences

The ViewDefault directive displays the report according to the user's preferences and privileges. Use the following syntax to display the report according to the user's settings:

`http://<webserver>/acweb/<reportserver>/<folder>/<file>?ViewDefault`

For information about the general syntax of directives, see "Directives syntax," earlier in this chapter.

If the user prefers viewing reports using the LRX, but does not have read permission on the report, ReportCast displays the report in DHTML format.

Search directives

You can build HTML forms for your users to use to specify search criteria. This section describes the directives that provide searching capabilities on DHTML reports. This section also explains how to build search URL's that return the requested search results.

For more information about performing searches on DHTML reports using the search request page, see "Searching DHTML reports" in Chapter 1, "Using ReportCast."

Requesting a search

The directives that support searching DHTML reports are:

- The searchReport directive. Report developers can embed searchReport directives in Actuate reports or specify them as hyperlinks in any HTML document. Users can also type searchReport directives in their web browsers. Users can bookmark searchReport URL directives.
- The requestSearch directive displays the search request page.

The searchReport directive

This section describes the searchReport directive.

The syntax is:

`http://<webserver>/acweb/<reportserver>/<folder>/
 <file>?searchReport[&format=<format>][&startingPoint=<start>][&hits=<hits>]
 >][&frameset=true|false]&<search list>`

<webserver>/acweb/<reportserver>/<folder>/<file>

Required. The URL for the DHTML report.

?

Required. Character indicating the start of a URL directive.

searchReport

Required. The URL search directive.

&format=<format>

Optional. Specifies the search results format. Possible formats are:

- DISPLAY, the standard DHTML display format. This is the default search results format.
- ANALYSIS, available if you have Actuate e.Analysis installed.
- CSV, comma separated values. Download only.
- TSV, tab separated values. Download only.

&startingPoint=<start>

Optional. Specifies the first result set data to return, beginning with the first result set. The default value, 0, returns the first result set.

&hits=<hits>

Optional. Specifies the total number of results to return.

&frameset=true|false

Optional. Specifies whether or not to reload the full frameset. The default value is true. The DHTML search ignores this parameter if format is DISPLAY. Developers can use this parameter to specify that only the search frameset should be reloaded. Users normally do not use this parameter.

&<search list>

Required. One or more search criteria, optionally followed by select criteria, separated by &'s. The syntax for <search list> is:

&<search criterion>[&<select criterion>][&<search criterion>] ...

The syntax for <search criterion> is:

<class>[.<variable>][:include | :exclude] [= "[value]"]

<class>

Required. The fully qualified name of a component class, such as the control component class. If you are creating a searchReport URL to embed in a report or web page, examine the report's .ROD in Actuate e.Report Designer or e.Report Designer Pro to determine the fully qualified name of the component class you wish to search on.

For example, to specify a search for Credit Rank while viewing the sample report BYFR.ROI, specify CustomerTitleFrame::TextControl8. The BYFR.ROI report is available in the Examples directory of your ERD or ERDPRO installation.

<variable>

Optional. The class variable on which to search. If you do not specify <variable>, the search uses the default value for the class.

<value>

Optional. The value or value expression to search for. If you do not specify <value>, the search uses the empty string, "".

Value expressions can include relational operators, logical operators, and metacharacters. For example, <, >, !, *.

If the value expression contains special characters (>, <, &, ;, @, #) or has one or more spaces at the beginning or end of the expression, enclose the expression in double quotes. For example, C3=">10" specifies a search for a string containing the characters ">10". The value expression C3=">10" does not specify a search for all C3 values greater than 10. ContentField=" 6* " specifies a search for a string containing the characters " 6* ". The value expression ContentField=" 6* " does not specify a search for all ContentField values containing the number 6 followed by any other characters.

:include|:exclude

Optional. The select modifier. The select modifier determines whether or not a specified select criterion, or component instance, is included in the search results. Specify :include to include the component instance in the search result. Specify :exclude to use the search criteria for the component class, but do not display the found instances.

For example, if you build a URL with the select modifier EmpSal:exclude=>0, the search uses the value EmpSal >0 as a select criterion, but does not return values for EmpSal in the search results.

The modifier :include is the default. The modifier is case insensitive.

The searchReport <select criterion> indicator

The optional indicator <select criterion> indicates whether the search includes the component instance in the search results. It differs from the select modifier :include|:exclude in that <select criterion> allows you to specify select-only criteria. Specifying select-only criteria causes the search to include selected components unconditionally in the search results set. The <select criterion> parameter also allows you to build custom search URL's using HTML forms.

The syntax for <select criterion> is:

```
<class>:select[=[<value>]]
```

<class>

Required. The fully qualified name of a component class. For example, OrderTitleFrame::OrderNumber.

:select

Optional. Tells the search to unconditionally include the component in the select results. The value of select overrides the value of the :include|:exclude search modifier.

<value>

Optional. The value of :select. Possible values are:

- TRUE
- FALSE

The values true and false are case insensitive. If you specify :select and <value> is not FALSE, the default value for :select is TRUE. For example, the following select criteria are identical:

```
C1:select
C1:select=true
```

Building a <select criteria> HTML form

Suppose that you want to create an HTML form to execute custom searches in the Customer report. You want to provide the ability to find all customers with a particular credit rank, and you decide to use a checkbox to allow your users to be able to decide whether or not to include the credit rank along with the customers' names and phone numbers.

The search criteria are:

```
CreditRank=*&CustomerName=*&PhoneNumber:select
```

Suppose that your web server machine name is Greenland, your e.Reporting Server is RServer, and the customer report is in the Customer folder. The HTML form is similar to the following:

```
<FORM NAME="emp_search" METHOD=POST
ACTION="http://Greenland/acweb/RServer/Customer/cust.roi?searchReport">
  <INPUT TYPE="hidden" NAME="format VALUE="DISPLAY">
  <INPUT TYPE="hidden" NAME="startingPoint" VALUE="true">
  <INPUT TYPE="hidden" NAME="frameset VALUE="true">
  <INPUT TYPE="text" NAME="CustomerName:include" VALUE="*">
  <INPUT TYPE="text" NAME="CreditRank:exclude" VALUE="*">
  <INPUT TYPE="checkbox" NAME="CreditRank:select" VALUE="TRUE"
CHECKED>Show Credit Rank
  <INPUT TYPE="hidden" NAME="PhoneNumber:select" VALUE="TRUE">
  <INPUT TYPE="submit" VALUE="Find...">
</FORM>
```

You need to use both the CreditRank:exclude and CreditRank:select because of the way that web browsers process form checkbox data. The previous example's search requires using CreditRank as part of the search criteria. But if

the CreditRank:select checkbox is not checked, web browsers ignore CreditRank:select.

For example, here is the query string when CreditRank:select is checked:

```
searchReport&CustomerName:include=*&CreditRank:exclude=*&CreditRank:select=TRUE
```

CreditRank:select=TRUE overrides CreditRank:exclude, so CreditRank is included in the search results.

Here is the query string when CreditRank:select is not checked:

```
searchReport&CustomerName:include=*&CreditRank:exclude=*
```

So the search proceeds with CreditRank=*, but CreditRank data is not included as part of the search results.

Working with searchReport

The following guidelines apply to working with the searchReport URL directive:

- The searchReport parameters format, startingPoint, hits, and frameset are keywords. If a component class has one of these keywords as its name, the search interprets the component class name as a search URL parameter.
- URL parameters are case-insensitive.
- The search uses only the last occurrence of a parameter, search criterion, or select criterion if you repeat them within a search URL.
- Place search parameters before search and select criteria.
- Enclose search values that contain spaces, and special character such as & and *, within double quotes.
- Full qualified component names that appear in different report sections must have unique names.
- If you specify the :include |:exclude search modifier and do not specify a select criterion, the search uses the search modifier to assemble the select list.
- If you do not specify the search modifier, :include is the default.

The requestSearch directive

The syntax is:

```
http://<webserver>/acweb/<reportserver>/<folder>/<file>?requestSearch
```

Users can access the search request page to specify search criteria for DHTML report searches:

```
http://<webserver>/acweb/<reportserver>/<folder>/<file>?requestSearch
```

For information about the general syntax of directives, see “Directives syntax,” earlier in this chapter.

Users display the search request page by clicking the Search link on the navigation bar.



Submitting a search

The `submitSearch` directive optionally submits a search request and returns the search results page. The search criteria are included in the search form data, contained in the URL.

Users generally submit search requests by filling out the search request page, then clicking the Search Now button.

Extracting search results

The `extractSearchResults` directive optionally submits a search request and returns the search results in a specified format. The search criteria are included in the search form data, contained in the URL.

Searches are case-sensitive.

File directives

This section describes the file directives that access report files on the e.Reporting Server.

Downloading report files

The syntax to use for downloading directives is as follows:

`http://<webserver>/acweb/<reportserver>/<folder>/<file>`

Send this directive to download an entire report file from the e.Reporting Server. If you configured the web browser to run the Actuate LRX or if you chose Live Report Extension as your report viewing preference, and you are downloading an Actuate report document (.ROI), ReportCast launches the Actuate LRX to display the report. If you chose Dynamic HTML (DHTML) as your report viewing preference, ReportCast does not download the report. The report displays in your web browser in DHTML format.

Downloading report document pages on demand

To view a report through the LRX, you can download an entire report document, or download only the pages you need. For large reports, the second

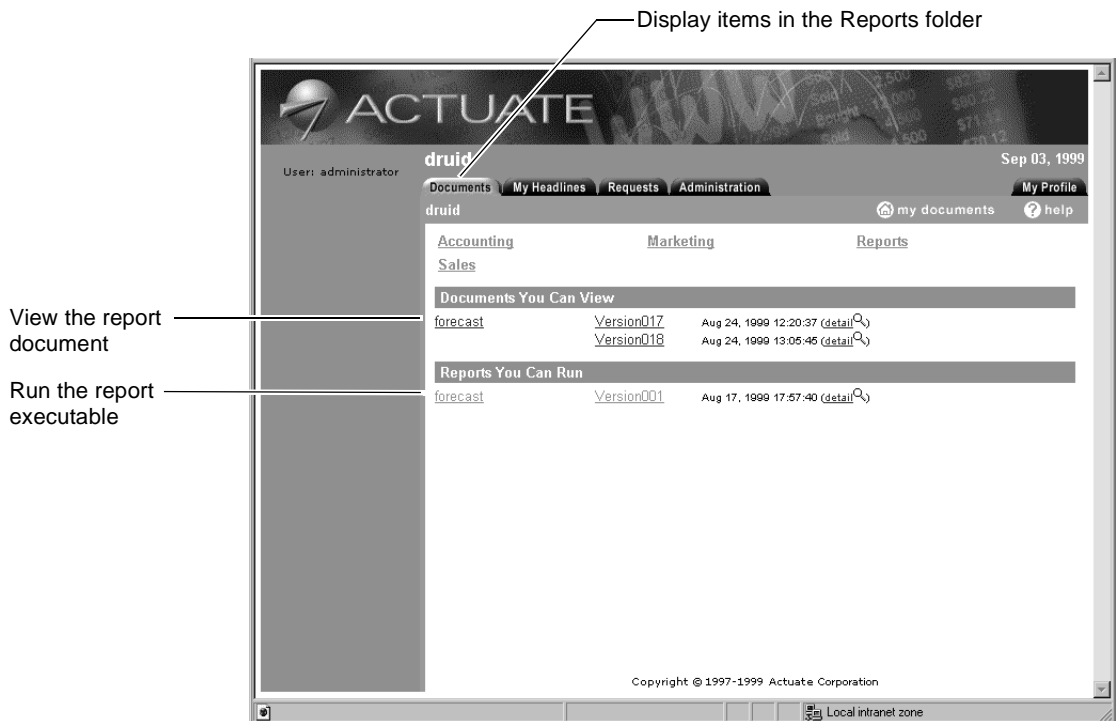
method, called demand paging, saves time and network resources. The syntax to use to download only select pages, is as follows:

`http://<webserver>/acweb/<reportserver>/<folder>/<file>?View`

This directive runs the Actuate LRX, which displays the report in a browser window.

You can use the view directive to view HTML reports (.ROW files). Demand paging does not apply to HTML reports. In HTML reports, ?View simply displays the entire report in the browser. Use the LRX's navigation buttons to go to particular pages in a report, or move to the report's first, next, previous, or last pages.

The view directive is associated with links in the Name column of the table that displays all the report files in a Report Encyclopedia folder. What happens when you use the link depends upon the type of document. For example, in this case, you can only view a document that is a report document but you can run a document that is a report executable. If you hold your cursor over the icon, the status bar displays information about the link.



Specifying file versions

When you issue a directive to download a file, you can also specify a specific version number. For example, the following directive downloads version 2 of DETAIL.ROI in the Reports folder on the Paradise e.Reporting Server:

`http://xanadu/acweb/paradise/reports/detail.roi;2`

If a folder contains multiple versions of a report and you do not specify a version number, ReportCast downloads the latest version. If you specify a version number that does not exist, ReportCast displays an error message.

Factory directives

This section describes the ReportCast factory directives. Factory directives generate new reports from report executables (.ROX) or report parameter values (.ROV). Factory directives also submit requests for execution.

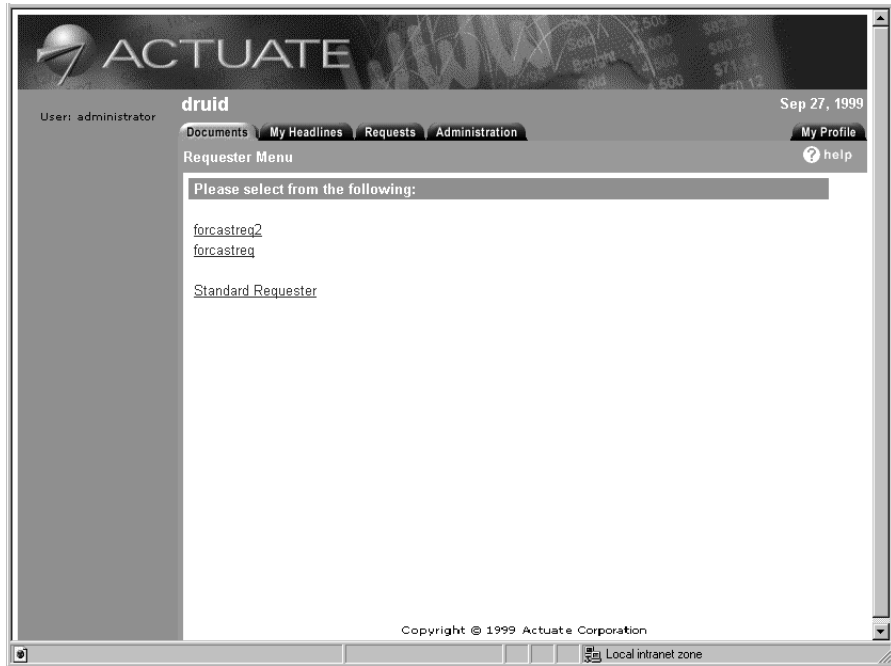
Using request

The syntax for the ?Request and ?StdRequest is as follows:

`http://<webserver>/acweb/<reportserver>/<folder>/<.rox | .roi | .rov |
.row>?Request | ?StdRequest`

The ?Request and ?StdRequest directives generate requester forms that run reports. The requester forms display in the web browser. Use those request forms to fill in request parameters.

Use the ?Request directive to generate custom requester forms. Use the ?StdRequest directive to generate the default requester form. If more than one requester form is available, ReportCast displays a list of the forms to choose from, as shown in the following illustration.



Custom requester forms make running reports simpler for your users. For example, suppose that your regional sales departments run a particular generic report on a regular basis. If you use a standard request, each time the sales staff runs the report, they have to fill in the region-specific information on the form. That information is always the same, so filling it in each time can be tedious. You can create a requester form for each region that supplies the appropriate parameter values for that region.

You can create as many custom requester forms as you need, but there must always be a default requester form for ReportCast to use. The type of requester form ReportCast generates depends on the type of file.

Report file type	Form
.ROX	Generates a blank requester form from the .ROX file parameter definitions.
.ROV	Generates a blank requester form using the .ROV file values as defaults.
.ROI	Looks for the .ROV file and uses it to generate a blank requester form. If ReportCast cannot find an .ROV file, it uses an .ROX file.

Report file type	Form
.ROW	Looks for the .ROV file and uses it to generate a blank requester form. If ReportCast cannot find an .ROV file, it uses an .ROX file.
.HTML	Generates a custom HTML requester form for a particular report, depending on a specific .ROX or .ROV file.

The following illustration shows the standard requester form for the Byfr report. The Byfr report is in your Actuate4\Devwb\Examples\FirstRpt directory.

The screenshot shows the Actuate druid web interface. At the top, the 'druid' logo is on the left, and the date 'Aug 30, 1999' is on the right. Below the logo, the user is identified as 'administrator'. A navigation bar contains links for 'Documents', 'My Headlines', 'Requests', and 'Administration'. A 'Submit Request' button is located on the left side of the page. The main content area is titled 'New Request' and contains several sections: 'Parameter Groups' with 'Output Parameters' (a 'Headline' text field) and 'Request' (a 'Wait for results' checkbox checked with '(when running now)'); 'Advanced' section with 'Schedule' (radio buttons for 'Right Now', 'Once', and 'Recurring', with 'Recurring' selected and a dropdown set to 'Every Day' and a time field set to '15:57:59'); 'Priority' (radio buttons for 'High (800)', 'Medium (500)', 'Low (200)', and 'Other (1-1000)', with 'Medium (500)' selected and a value of '500' in a text field); 'Version' (radio buttons for 'Create new version' and 'Overwrite existing version', with 'Create new version' selected, and a 'Keep only the latest' checkbox and 'version(s)' text field); 'Archive Policy' (radio buttons for 'Use the archive policy for the distribution folder(s)', 'Delete objects older than', and 'Delete objects on', with 'Use the archive policy for the distribution folder(s)' selected); and 'Notify Channels' (a 'News' checkbox). At the bottom, there is a 'Submit Request' button and a copyright notice: 'Copyright © 1997-1999 Actuate Corporation'.

The following illustration shows the VIP custom requester form for the Byfr report. The VIP requester does not wait for the report to finish, runs the report

at the highest priority level by default, and overwrites the existing report version rather than creating new versions each time.

The screenshot shows the ACTUATE druid web interface. The header includes the ACTUATE logo, the user 'administrator', the date 'Aug 30, 1999', and navigation tabs for 'Documents', 'My Headlines', 'Requests', and 'Administration'. A 'Submit Request' button is on the left, and a 'My Profile' link with a help icon is on the right. The main content area is titled 'New Request' and contains several sections: 'Parameter Groups' with an 'Output Parameters' section for a 'Headline' (a text input field) and a 'Request' section with a 'Wait for results' checkbox (labeled '(when running now)'). The 'Advanced' section includes a 'Schedule' part with radio buttons for 'Right Now', 'Once' (with date and time inputs), and 'Recurring' (with a dropdown for frequency and time input). The 'Priority' section has radio buttons for 'High (800)', 'Medium (500)', 'Low (200)', and 'Other (1-1000)', with a text input field set to '500'. The 'Version' section has radio buttons for 'Create new version' (with a 'Keep only the latest' checkbox and version input) and 'Overwrite existing version' (with 'Version Name' and 'Output Name' inputs). The 'Archive Policy' section has radio buttons for 'Use the archive policy for the distribution folder(s)', 'Delete objects older than' (with days and hours inputs), and 'Delete objects on' (with date and time inputs), plus a checkbox for 'Archive items before deletion'. The 'Notify Channels' section has a checkbox for 'News'. A second 'Submit Request' button is at the bottom left. The footer contains the copyright notice 'Copyright © 1997-1999 Actuate Corporation'.

ACTUATE

User: administrator Aug 30, 1999

Documents My Headlines Requests Administration My Profile ? help

Submit Request

New Request

Parameter Groups

Output Parameters

Headline (String)

Request

Wait for results ☐ (when running now)

Advanced

Schedule

☒ Right Now

☐ Once (mm/dd/yyyy) at (hh:mm:ss)

☐ Recurring at (hh:mm:ss)

Priority

☒ High (800) ☐ Medium (500) ☐ Low (200) ☐ Other (1-1000)

Version

☐ Create new version ☐ Keep only the latest version(s)

☒ Overwrite existing version

Version Name:

Output Name:

Archive Policy

☒ Use the archive policy for the distribution folder(s)

☐ Delete objects older than days hours

☐ Delete objects on (mm/dd/yyyy) at (hh:mm:ss)

☐ Archive items before deletion (only applies if not using archive policy for distribution folder)

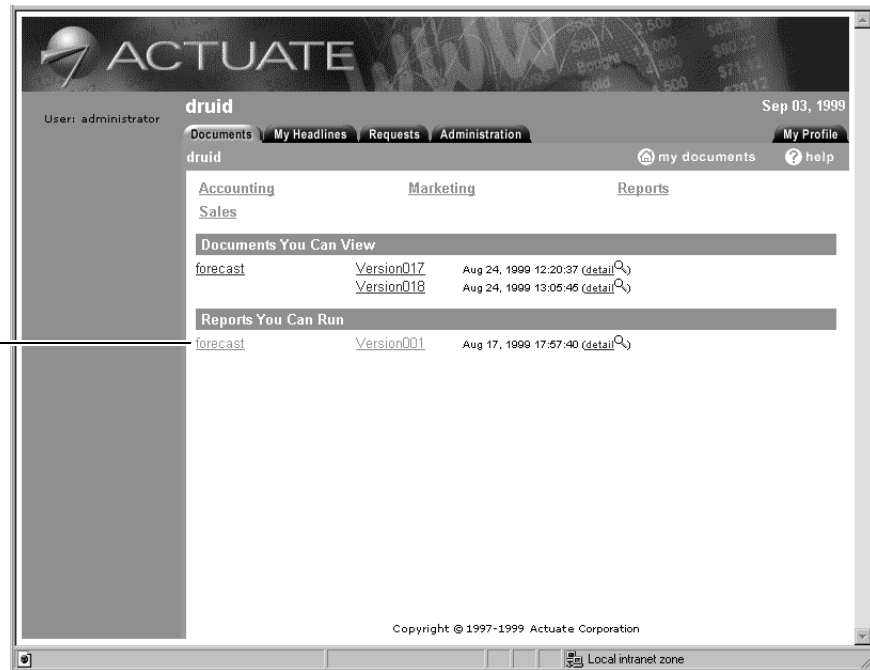
Notify Channels

☐ News

Submit Request

Copyright © 1997-1999 Actuate Corporation

Clicking this link generates the ?Request directive shown on the status line.



Using submit

The ?Submit directive submits requests for execution in the e.Reporting Server. Users typically submit requests by clicking a Submit button in a requester form, but you can also generate requests and submit the parameters directly in a submit URL.

The syntax for the ?Submit directive is as follows:

`http://<webserver>/acweb/<reportserver>/<folder>/<rox>?Submit`

Using a submit directive

Choosing the Submit Request button at the bottom of the form starts the report generation process. When you choose Submit Request, the form submits the parameter values you specified to ReportCast. ReportCast then issues the following directive, which displays a page with information about the report generation request:

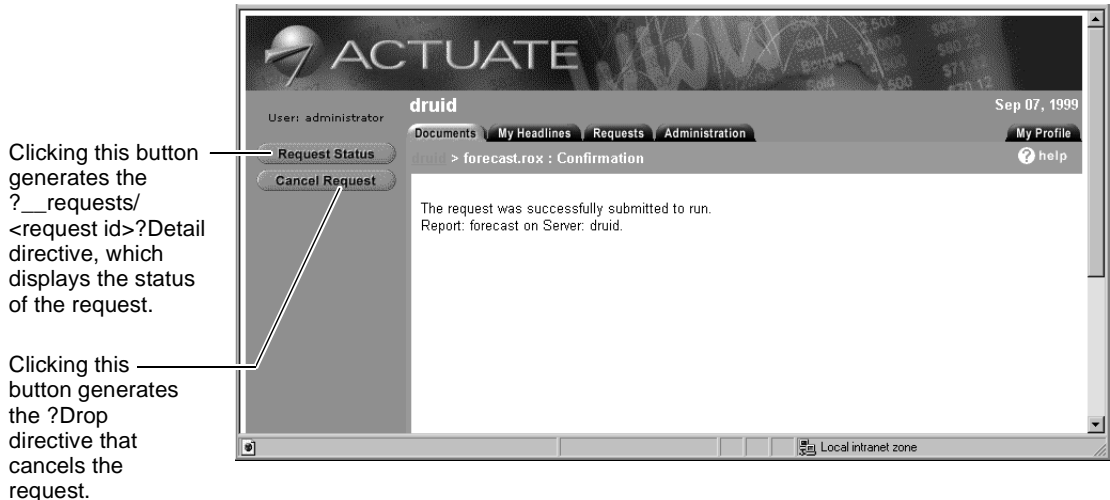
`http://<webserver>/acweb/<reportserver>/<folder>/<.rov> | .<rox>?Submit`

The e.Reporting Server places the resulting .ROX in the user's home folder, if the user has a home folder, or in the same folder as the .ROI.

Always include the ?Submit directive when you create custom requester forms. For example, the following directive sets up a submit button:

```
<form method="POST" action="$(REPORTURL)?Submit">
  <input type="image" value="Submit Request" img src="MySubmitBtn.gif">
```

This directive is typically associated with a button or link. The following illustration shows the results of the ?Submit directive.



Using a submit URL

You can submit report generation requests directly to ReportCast using submit URLs. The syntax is:

```
<reportserver path>/<folder>/<rox> | <rov>
  ?Submit&variable=value[&variable=value...]
```

<reportserver path>

The HTTP path for your report server.

<folder>

The e.Reporting Server folder that contains the report executable from which to generate the report.

<rox> | <rov>

The report executable or parameter values file from which to generate the report.

?Submit

The submit directive.

&

The HTTP separator that separates the parameter values the e.Reporting Server uses to generate the report.

variable=value

The variable=value pairs define the report parameters and the values to use to generate the report. Values may never include quotes. Use %20 to represent spaces in any string values you define for the report. Variables are case-sensitive. You must specify date and time values in the format appropriate for your locale.

The following table lists some standard submit parameter variables. You define report parameters as part of your report design in the e.Report Designer Professional. Report parameters are case-sensitive. Refer to them exactly as you define them in e.Report Designer Professional.

Variable	Description
__accessToGrant	The type of access to grant automatically to those roles that have permission to view the report. Permission is granted by a combination of the __channels and __exclude values. If the output is a secure DHTML report, grant Secure Read access. Otherwise, grant Read access.
BundleRox	True to bundle the data along with the generated .ROI. False otherwise. False is the default value.
Headline	A descriptive tagline for a report. Displays on the Channel Contents page. Use the character string %20 to represent spaces in the headline string.
__scheduleType	Required. The type of schedule: immediate, once, or recurring. Immediate is the default.
__wait	Optional. Wait while the report generates. Can be used only if the report's __scheduleType is immediate.
__onceDate	Required for once schedules. The date on which to run the report, for reports with __scheduleTypes of once. Must be in the appropriate format for your locale. For example, the format for the U.S. (enu) locale is mm/dd/yy. The current date is the default.

Variable	Description
<code>--_onceTime</code>	<p>Required for once schedules.</p> <p>The time at which to run the report, for reports with <code>--_scheduleTypes</code> of once. Must be in the appropriate format for your locale. For example, the format for the US (enu) locale is hh:mm:ss. The current time is the default.</p>
<code>--_schedulePeriod</code>	<p>Required for recurring schedules.</p> <p>How often to run the report, and on which day(s). Choose a day of the week.</p> <p><code>--_schedulePeriod</code> values are Every Day, Weekdays, Mondays, Tuesdays, Wednesdays, Thursdays, Fridays, Saturdays, Sundays, First Day of the Month, Last Day of the Month. All values are case-sensitive.</p> <p>Every Day or Weekdays. Set only if <code>--_scheduleType</code> is recurring.</p>
<code>--_recurringTime</code>	<p>Required for recurring schedules.</p> <p>The time at which to run the report. Set only if reports <code>--_scheduleTypes</code> is recurring.</p> <p>Must be in the appropriate format for your locale. For example, the format for the U.S. (enu) locale is hh:mm:ss.</p>
<code>--_priority</code>	<p>Optional.</p> <p>The importance that generating this report has in the e.Reporting Server priority.</p> <p>An integer value between 1 and 1000, or Other. 200 is low, 500 is medium, 800 is high. 500 is the default.</p> <p>To set <code>--_priority</code>, use a radio button group in the HTML requester form.</p>
<code>--_priorityValue</code>	<p>Optional.</p> <p>Indicates request priority, if the value of:</p> <ul style="list-style-type: none"> <code>--_priority</code> is not Other <code>--_priorityValue</code> is an integer between 1 and 1000
<code>--_overwrite</code>	<p>Optional.</p> <p>New to create a new version of this report. New is the default.</p> <p>Old otherwise.</p>

Variable	Description
<code>__versionname</code>	Optional. A string value for the new version name of this report.
<code>__outputname</code>	Optional. Name of the output. ROI file for this report. The default is <code><report>.roi</code> , where <code><report></code> is the name of the .ROX file.
<code>__channels</code>	Optional. Name of a channel to notify of this request. You can notify more than one channel.
<code>__exclude</code>	Optional. The name of a role that must not have access to the report. You can exclude more than one role.
<code>__groups</code>	Optional. The name of the group to notify of this request. You can notify more than one group.
<code>__users</code>	Optional. The name of the user to notify of this request. You can notify more than one user.
<code>__wait</code>	Optional. Indicate whether or not to wait for the report to complete, then immediately display the first page. Use only if <code>__scheduleType</code> is set to "immediate". Default is to wait for the report to complete.
<code>__limit</code>	Optional. Indicate whether or not to limit the number of versions of the output files for the current request. Set <code>__wait</code> to "limit" to limit the number of versions. Any other value means that the number of versions is unlimited.
<code>__limitNumber</code>	Optional. The number of versions to which to limit the output files for the current request. Use only if <code>__limit</code> is set to "limit". <code>__limitNumber</code> can be any positive number.

Variable	Description
<code>__archivePolicy</code>	Optional. The archive policy to implement for the objects created as output for the current request. Values are folder, age, and date. Set “folder” to use the archive policy that is already set for the folders to which the output is distributed. Set “age” to delete objects older than a specific time period. Set “date” to delete objects on a specific date.
<code>__ageDays</code>	Optional. Use with <code>__ageHours</code> to determine how long output objects exist before they are deleted. Use only if <code>__archivePolicy</code> is set to “age”. <code>__ageDays</code> can be any positive number.
<code>__ageHours</code>	Optional. Use with <code>__ageDays</code> to determine how long output objects exist before they are deleted. Use only if <code>__archivePolicy</code> is set to “age”. <code>__ageHours</code> can be any positive number.
<code>__dateToDelete</code>	Optional. The date on which to delete the output objects of the current request. Use only if <code>__archivePolicy</code> is set to “date”. <code>__dateToDelete</code> must be a date in a locale-specific format. The default format is “mm/dd/yyyy”.
<code>__timeToDelete</code>	Optional. The time on the date specified by <code>__dateToDelete</code> on which to delete the output objects of the current request. Use only if <code>__archivePolicy</code> is set to “date”. <code>__timeToDelete</code> must be a time in a locale-specific format. The default format is “hh:mm:ss”.
<code>__archiveBeforeDelete</code>	Optional. Indicate whether or not to archive the output objects of the current request before deleting them, according to <code>__archivePolicy</code> ’s setting. This has no effect if <code>__archivePolicy</code> is set to “folder”. Set to TRUE to archive objects before deleting them. The default value is FALSE.

For example:

```
http://mustique/acweb/caligari/Test/
MoSales.rox?Submit&Headline=Monthly%20Sales%20Report&__scheduleTy
pe=recurring&__schedulePeriod=Last%20Day%20of%20the%20Month&__re
curringTime=01:30:00&__users=Admin&__groups=Sales&__groups=Mktg
```

This example submits the report executable MOSALES.ROX to the e.Reporting Server, Caligari. Its headline is Monthly Sales Report. It is on a recurring schedule to run on the last day of each month at 1:30 am. The user Admin and groups Sales and Mktg will be notified of the report run.

Using status directives

Status directives retrieve status about specific report-generation requests, and about all report-generation requests handled by the factory on the specified e.Reporting Server.

Getting status of a report-generation request

The syntax to initiate a request for status of a report-generation request is as follows:

```
http://<webserver>/acweb/<reportserver>/__requests/<request id>?detail
```

This directive displays a page that shows detailed status information about the report-generation request. Typically, you do not enter this URL directly. Instead, you click a link that generates this directive.

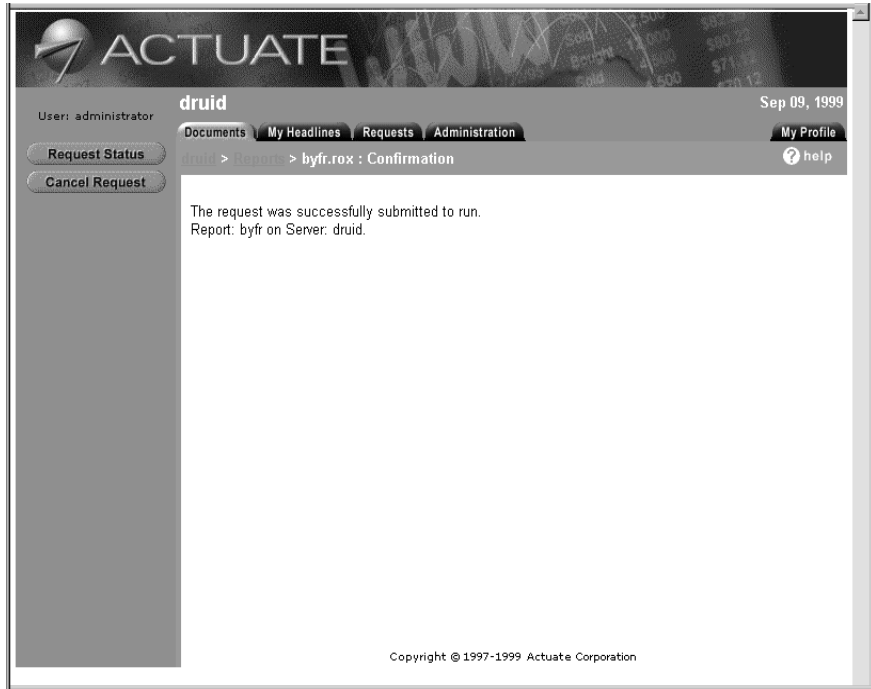
Cancelling a report-generation status request

The syntax to cancel a request for status of a report-generation request is as follows:

```
http://<webserver>/acweb/<reportserver>/__requests/<request id>?drop
```

This directive cancels the request for status of a report-generation request just submitted. Typically, you do not enter this URL directly. Instead, you click a link that generates this directive, such as the Cancel Request button on the Request Successfully Submitted page.

The following illustration shows a status page for a report-generation request. At the bottom of that page are links you can select to cancel the report-generation job, get more status information, or view the report when it is generated.



Deleting status for a request

The syntax to delete the request for status of a report-generation request is as follows:

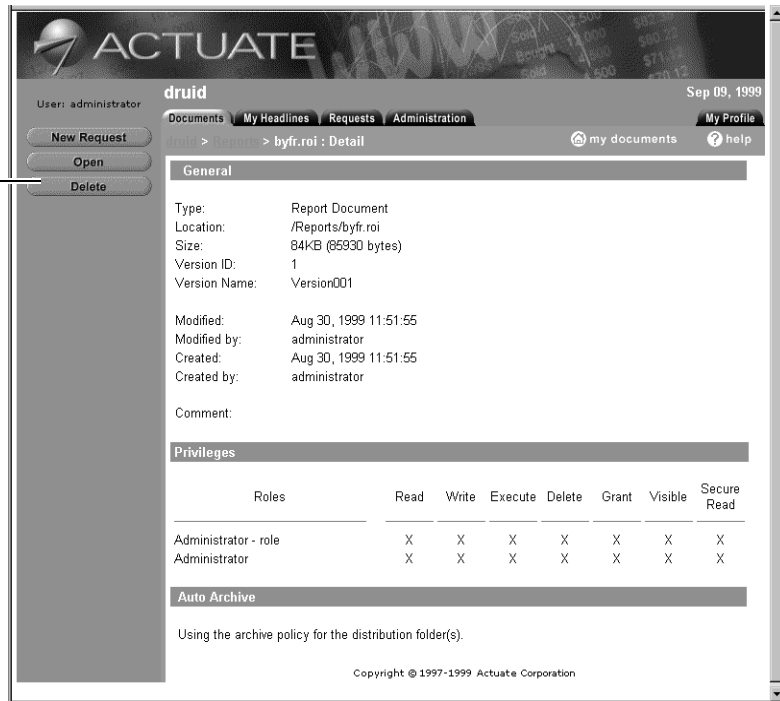
`http://<webserver>/acweb/<reportserver>/_ _requests/<request id>/
?DeleteStatus`

This directive deletes any status the current user has for the request specified by <request id>. ReportCast creates status information automatically as a report generates.

For example, if you generate a particular report or set of reports, ReportCast creates status for each of them. The status is available from a report's details page. When you view the list of completed requests on your report server, you access the status by clicking on the report's Details link. The Details page contains a Delete Notice button. Click that button to delete the status for that report.

The following illustration shows a report's Details page and the Delete Notice button.

Clicking this button generates the ?Delete Status directive.



Using information directives

Information directives provide detail about files, folders, and requests. The syntax for the ?Detail directive is as follows:

`http://<webserver>/acweb/<reportserver>/<folder>/<file>?Detail`

?Detail provides precise information about a Report Encyclopedia object. The information provided is usually too extensive and complex to display in ReportCast's various summary views. The previous illustration of Active Requests Details on Mustique shows detail information for a .ROX file.

Information includes:

- All operations allowed on that object
- User privileges for accessing the object

ReportCast uses a format that is similar to the Navigator's property pages format to display the information. For more information about the Navigator, see Chapter 1, "Understanding the Administrator Desktop," in *Administering the Report Encyclopedia*.

Getting information about factory requests

You can get information about factory requests by using the status directive URLs. The syntax is:

`http://<webserver>/acweb/<reportserver>/<status directive>`

The following table lists the status directives.

Directive	Displays ...
<code>--_scheduled</code>	Factory requests currently scheduled
<code>--_active</code>	Factory requests currently active
<code>--_completed</code>	Factory requests completed

These directives are associated with the Scheduled, Active, and Completed links on the HTML page that displays the contents of a folder.

About Channel directives

Use channel directives to administer, view the contents of, subscribe to, and unsubscribe from ReportCast channels. For more information about channel directives, see “Working with ReportCast channels,” in Chapter 1, “Using ReportCast”.

The syntax of channel directives is:

`http://<webserver>/acweb/<reportserver>/<channel directive>`

The following table lists the channel directives.

Directive	Description
<code>--_channels?admin</code>	Displays the channel administration page
<code>--_completed</code>	Displays the list of channels to which the user is subscribed
<code>--_channels/<channel></code>	Lists channel contents <channel>
<code>--_channels?retrieveForm</code>	Displays the subscription form for available channels
<code>--_channels?submitForm</code>	Submits the subscription information

Directive	Description
<code>__channels?retrieveCreation</code>	Displays a form to create a new channel Available to administrators only
<code>__channels?submitCreation</code>	Submits the new channel creation form Available to administrators only
<code>__channels/channel?drop</code>	Removes an existing channel Available to administrators only
<code>__channels/channel?retrieveDetails</code>	Displays channel attributes modification form Available to administrators only
<code>__channels/channel?submitDetails</code>	Submits the channel details modification form Available to administrators only

Listing ReportCast channels

The syntax of the channel directive is:

`http://<webserver>/acweb/<reportserver>__completed`

This directive lists all available ReportCast channels. Channels to which the user already subscribes display in a different color than those to which the user does not subscribe. Users with administrator privileges can customize this list using the ReportCast templates.

Administering ReportCast channels

The syntax of the available administrative directives are:

`http://<webserver>/acweb/<reportserver>/__channels?admin`

`http://<webserver>/acweb/<reportserver>/__channels?retrieveCreation`

`http://<webserver>/acweb/<reportserver>/__channels?submitCreation`

`http://<webserver>/acweb/<reportserver>/__channels/
<channel>?retrieveDetails`

`http://<webserver>/acweb/<reportserver>/__channels/
<channel>?submitDetails`

`http://<webserver>/acweb/<reportserver>/__channels/<channel>?drop`

These directives allow users with administrator privileges to display the channel administration page, add new channels, modify existing channels, and drop existing channels.

For example, to display the channel administration page for a ReportCast service running on Mustique's web server, use the following URL:

`http://mustique/acweb/mustique/_ __channels?admin`

To drop a channel called `prodmkt`, use the following URL:

`http://mustique/acweb/mustique/_ __channels/prodmkt?drop`

Displaying channel contents

The syntax of the directive is:

`http://<webserver>/acweb/<reportserver>/_ __channels/<channel>`

This directive displays the channel contents `<channel>`. The contents are a list of completed requests. Some channels may have restricted access. If a particular user is not allowed to access a particular channel, an error message displays in the web browser.

Creating channels using channel URLs

You can create new channels or change existing channels directly using channel URLs. The syntax is:

`<reportserver path>/
__channels?submitCreation&variable=value[&variable=value...]`

or:

`<reportserver path>/__channels/
<channel>?submitDetails&variable=value[&variable=value...]`

<reportserver path>

The HTTP path for your e.Reporting Server.

__channels?submitCreation

The channels directive followed by the `submitCreation` command.

&

The URL separator character. Separates the parameter values that the e.Reporting Server uses to create or modify the channel.

variable=value

Specifies the channel parameters and the values to use to create or modify the channel properties. Values can never include quotes. Use `%20` to represent spaces in any string values you define for the channel, such as the channel

description. Specify numbers and times in the format appropriate for your locale.

The following table lists the channel parameter variables.

Variable	Description
__ChannelName	Required for channel creation. Names the new channel or the channel to modify.
__Comment	Optional. Describes the channel.
__Expiration	Required. Number of days before item expires from the channel.
__LargeIconURL	Optional. URL to a 32x32-pixel icon that represents the channel.
__PollingInterval	Required. Refresh frequency, in seconds, of the channel contents page.
__Roles	Optional. Names a role that has permission to subscribe to this channel. More than one role can subscribe to a channel.
__SmallIconURL	Optional. URL to a 20x20 pixel icon that represents the channel.

For example:

```
http://mustique/acweb/caligari/  
__channels?submitCreation&__ChannelName=News&__Comment=News%20  
you%20can%20use.&__Expiration=3&__PollingInterval=30&__Roles=All
```

This example creates a new channel called News on the report server Caligari. The channel description is News you can use. Items expire out of the channel when they are three days old. The channel checks for new items every 30 seconds. Users with the role All can subscribe to the channel.

Subscribing to channels using channel URLs

You can subscribe to channels directly using channel URLs. The syntax is:

```
<reportserver path>/  
__channels?submitForm&variable=value[&variable=value...]
```

<reportserver path>

The HTTP path for your e.Reporting Server.

__channels?submitForm

The channels directive, followed by the submitForm command.

&

The URL separator character. This character separates the parameter values that the e.Reporting Server uses to create or modify the channel.

variable=value

Specifies channel parameters and the values used to subscribe to the channel

The following table lists the channel subscription variables.

Variable	Description
channels	Optional. Names of the channel to which to subscribe. You can subscribe to more than one channel.

For example:

```
http://mustique/acweb/caligari/  
__channels?submitForm&channels=News&channels=Sales
```

This example subscribes the user to the News and Sales channels on the Caligari report server.

Examples of ReportCast directives

The following section provides several examples of directives you enter in the browser's Location or Address box.

The following directive displays the Paradise e.Reporting Server contents. Actserv is the name of the machine on which the web server and ReportCast are installed:

```
http://actserv/acweb/paradise/
```

The following directive displays the contents, sorted by type, of the Reports folder in the Paradise e.Reporting Server:

```
http://actserv/acweb/paradise/reports/?sort=Type
```

The following directive displays only the report files excluding subfolders, sorted by type, in the Reports folder on the Paradise e.Reporting Server:

```
http://actserv/acweb/paradise/reports/?sort=Type&files
```

The following directive displays the request form for running the
DETAIL.ROX report:

<http://actserv/acweb/paradise/reports/detail.rox?Request>

The following directive submits the request form for running the
DETAIL.ROX report:

<http://actserv/acweb/paradise/reports/detail.rox?Submit>

The following directive displays all the scheduled factory requests on the
Paradise e.Reporting Server:

http://actserv/acweb/paradise/_ _scheduled

The following directive displays a list of all the channels to which the user is
subscribed. This same directive displays the contents of the user's completed
folder:

http://actserv/acweb/paradise/_ _completed

The following directive displays the contents of channel Reports. The contents
take the form of a list of the completed requests available for viewing:

http://actserv/acweb/paradise/_ _channels/Reports

The following directive displays the channel subscription form:

http://actserv/acweb/paradise/_ _channels?retrieveDetails

The following directive submits the channel subscription form:

http://actserv/acweb/paradise/_ _channels?submitDetails

ReportCast scripting language reference

This chapter contains the following topics:

- About the ReportCast scripting language
- ReportCast scripting language reference
- ReportCast scripting language variables

About the ReportCast scripting language

ReportCast scripting language commands tell ReportCast how to fill in information when it generates web pages from the template files. ReportCast templates determine the design and presentation of generated web pages. The templates contain a mixture of normal HTML tags and Actuate extensions. The Actuate extensions are the ReportCast scripting language.

The ReportCast scripting language allows you to create custom reporting applications and integrate them with web applications. You can create sets of templates for different audiences and uses, such as:

- Custom templates for specific departments, such as Sales or Accounting
- Internationalized templates for generating web pages in a particular language, such as Japanese, French, or English

The elements of the ReportCast scripting language are:

- **Commands.** Commands tell ReportCast how to insert information when it generates web pages. Commands are case insensitive and must appear in an HTML comment that starts with #Actuate. Each scripting language command must start at the beginning of the line in the template file. Do not place spaces or tabs before the HTML comment containing the scripting language command.
- **Variables.** ReportCast substitutes variable names with real data when it generates web pages. Variable names are case insensitive and may appear anywhere in a line.

ReportCast scripting language reference

This section presents an alphabetical listing of the ReportCast scripting language commands and their syntax. Each command entry includes a general description of the command and a summary of its options.

Scripting language commands are of the following types:

- **Sequential.** These are commands that are executed in sequence, from beginning to end, as ReportCast encounters them in a template.
- **Iteration.** These are commands that loop through lists of objects. Objects include files, folders, properties, and requests. Iteration commands provide some of the same features as e.Report Designer Professional. You can specify the equivalents of Before, After, Content, and IfEmpty sections of reports.

- Selection. These commands provide structures for case statements and conditional statements.

Scripting language commands use the following syntax:

```
<!-- #Actuate <directive> [options] -->
```

#Actuate

- Required.
- Tells ReportCast that this is a command.

<directive>

- Required.
- Tells ReportCast what to do.

[options]

- Optional.
- Lists one or more options for the control directive.

The following table lists the scripting language commands.

Command	Description
List	Iterate through the contents of a list. This command is the most complicated. ReportCast provides many types of lists.
If	Conditionally includes a particular HTML fragment.
Include	Includes another scripting language file.
With	Works with a specified web object.
Set	Sets the value of a scripting language variable.

List command

Loop through the contents of a list.

Syntax

```
list keyword [options]
```

```
    ifempty
    before
```

content
after
end list

Description

List command provides the ability to loop through a set of lists defined by ReportCast. The ifempty, before, content, and after sections are optional. You can define sections of lists using these optional keywords. Lists begin with list keyword and end with end list. You can define sections of lists using the optional ifempty, before, content, and after keywords. If they are not present, anything between the list and end list keywords is considered to be the content section. Each section begins with its keyword, and ends at the next keyword.

The following table describes each list element.

List element	Description
keyword	Required. Specifies the type of list to process.
ifempty	Optional. Specifies an action to take if the list is empty.
before	Optional. Specifies content to appear before the main list section. Similar to e.Report Designer Professional's Before slot.
content	Optional. Specifies content to appear in the main list section.
after	Optional. Specifies content to appear after the main list section. Similar to e.Report Designer Professional's After slot.
end list	Required. Specifies the end of the list.

List keywords

List keywords specify the type of list to process. The next sections describe the List command's keywords in more detail. The following table describes the List keywords.

Keyword	Description
Files	Lists the files in the current Report Encyclopedia folder.
Versions	Lists the versions, if any, for the current file.
Folders	Lists the folders in the current Report Encyclopedia folder.
Items	Lists the folders and files in the current Report Encyclopedia folder.
RequestForms	Lists the available custom and standard request pages that run the report associated with the current .ROI, .ROW, .ROX, or .ROV file.
Requests	Lists all requests.
ParamGroups	Lists all parameter groups for a request.
Parameters	Lists all parameters for a request or group.
Values	Lists the values for a submitted request.
Status	Lists the status items for a request.
Members	Lists all members on the current e.Reporting Server. Members include all users, roles, and groups.
Grants	Lists all members that have explicitly been granted permissions for the current item.
Privileges	Lists the privileges each user and role has for the current item.
Notifications	Lists the users and groups to notify when the current request completes.
Schedules	Lists the string descriptions of schedules for the current request.
PrinterProps	Lists the properties of the printer being used for the current request.
Channels	Lists the channels available on the report server or the channels to which the current user is subscribed.

Keyword	Description
ReportServers	Lists the report servers to which ReportCast has open connections.
Expirationpolicies	Lists the expiration policies for the current file.
Pathlink	Lists the path links that apply to the current folder, file, or .ROD.
Tocitems	Lists the table of contents items to display.
Criteria	Lists the search criteria that were submitted as part of a search request.
Resulthits	Lists the search result hits returned by the view process.
Resultfields	Lists the fields that make up the search results.
Viewpageformats	Lists the supported formats for displaying DHTML report pages.
Searchformats	Lists the supported formats for presenting the results of search operations.

List files command

Lists the objects in the current Report Encyclopedia folder.

Syntax

list files [(type [, type...])] [sort by field [desc]]

Description

List files lists Report Encyclopedia objects in the current encyclopedia folder. You can list all files, or just files of a particular type. You can specify the sort order for the files, or use the sort order specified in the URL directive that generates the file list web page. This command lists only the files in the current folder. It does not list the contents of other folders within the current folder.

Example

The following example shows how to list the files in a Report Encyclopedia folder:

```
<!-- Start the list -->
<!-- #Actuate list files -->
```

```

<!-- Define the list heading -->
<!-- #Actuate before -->
<TABLE>
<TR>
<TH>Name</TH> <TH>Owner</TH> <TH>Date</TH>
<!-- Here is the main list content -->
<!-- #Actuate content -->
<TR>
<TD> $(FileName) </TD>
<TD>$(OwnerName)</TD>
<TD>$(Created)</TD>
</TR>
<!-- Now, define the end of the list -->
</TABLE>
<!-- Here's what to do if the list is empty -->
<!-- #Actuate ifempty -->
This folder does not contain any report items.
<P>
<!-- #Actuate end list -->

```

List files type clause

Specify the kinds of files to list. File types are indicated by the suffix of the file in the Report Encyclopedia, for example, .ROW, .ROI, .ROX. Types are case insensitive. You can specify wildcard characters in the type clause.

Wildcard	Matches
?	Any one character
*	Any number of characters
...	All files not previously matched by a preceding pattern

Example

The following example lists all the report executables in the current folder:

```
<-- #Actuate list files (ROX)
```

List files sort by clause

Specifies how to sort the files in the list.

Description

Takes the sort order from the URL directive, by default. The default sort direction is ascending. You can override the default sort order by specifying the sort by clause in your list files command. Use the desc keyword to change

the direction of the sort to descending. You can sort by only one field at a time. ReportCast cannot sort by multiple fields.

The following table lists the available sort fields.

Sort field	Description
Name	Sorts by file name and version number.
Type	Sorts by file type.
Owner	Sorts by file owner name.
Date	Sorts by file creation date.
Size	Sorts by file size.

Example

The following example lists all report executables, sorted by creation date, in ascending order:

```
<!-- #Actuate list files (ROX) sort by date
```

List versions command

List the available versions of the current file.

Syntax

list versions

Description

Each time the report runs, the e.Reporting Server creates a new version of the report file unless you specify otherwise. If a user attempts to view a version of a file in Dynamic HTML (DHTML) that is no longer available, list versions lists the available versions of that file.

List versions is available from:

- The file versions page, named FILEVERSIONS.ACHTML
- The file details page, named FILEDETAIL.ACHTML

Example

```
<!-- #Actuate list versions
```

List folders command

List all the folders in the current Report Encyclopedia folder.

Syntax

```
list folders [sort by field [desc]]
```

Description

List folders lists all folders in the current Report Encyclopedia folder. You can sort the list by specific fields. You can also specify whether the sort should be ascending or descending. The default sort order is ascending.

The following example lists all folders in the current Report Encyclopedia folder, sorted by owner, in descending order:

```
<--! #Actuate list folders sort by owner desc -->
```

List folders is available from the folder listing page, named FOLDERLIST.ACHTML.

See also

List files sort by clause.

List items command

Lists the contents of the current Report Encyclopedia folder.

Syntax

```
list items [sort by field [desc]]
```

Example

The following example lists all the items in the current Report Encyclopedia folder, using the default sort order. The default sort order is ascending:

```
<--! #Actuate list items -->
```

List items is available from the folder listing page, named FOLDERLIST.ACHTML.

See also

List files sort by clause.

List requestforms command

Lists the available requester forms.

Syntax

list requestforms

Description

Lists the requester forms available to run the report specified by the URL directive or associated with the file name in the URL directive. List requestforms is available only from ReportCast's Requester Menu page.

List requestforms is available from the requester menu page, named REQUESTERMENU.ACHTML.

List requests

Lists the requests on the current report server or for the current user.

Syntax

list requests

list {active | scheduled | completed} requests [sort by {name | date | status}]

Description

ReportCast uses the list requests command to generate a list of requests when it receives a URL directive for the __active, __completed, or __scheduled request folders. List requests is available only from ReportCast's Request Folder page.

The first form lists all requests on the current e.Reporting Server.

The second form lists requests for the current user. Use this form to create a web page other than a request folder page, for example, to display the user's completed requests on an e.Reporting Server's main page, so that their completed reports are immediately available.

List requests is available from the request list page, named REQUESTLIST.ACHTML.

List paramgroups, list parameters commands

Use these commands to create custom requester forms.

Syntax

list paramgroups

list parameters

Description


List paramgroups and list parameters provide a requester template for generating a custom requester form.

When users send a request to run a particular report, ReportCast generates a requester form. The requester form lists the available report parameters and allows users to fill in the parameters they are interested in. For example, users can run a standard sales report, filling in parameters such as region, sales office, quarter, and so on.

If many users from particular regions or departments plan to run the standard report regularly, you can provide custom requester forms that provide default values for parameters the users would otherwise have to fill in by hand. For example, you can create a requester form for the Western Region sales group, another for the Eastern Region sales group, and still other requester forms for the European, Asian, and South American sales groups.

If a particular parameter group name is empty—the value is ""—its parameters were defined at the top level in the Developers WorkBench Parameters dialog box. They do not really belong to a particular parameter group.

The following illustration shows a standard requester form for a report. The form lists all the parameters required for the report, including channels to notify when the report finishes running. The parameters are listed by group.


ACTUATE

User: administrator
 Aug 30, 1999

[Documents](#)
[My Headlines](#)
[Requests](#)
[Administration](#)
[My Profile](#)
[? help](#)

[Submit Request](#)

New Request

Parameter Groups

Output Parameters

Headline (String)

Request

Wait for results ☒ (when running now)

Advanced

Schedule

☒ Right Now

☐ Once (mm/dd/yyyy) at (hh:mm:ss)

☐ Recurring at (hh:mm:ss)

Priority

☐ High (800)
 ☒ Medium (500)
 ☐ Low (200)
 ☐ Other (1-1000)

Version

☒ Create new version ☐ Keep only the latest version(s)

☐ Overwrite existing version

Version Name:

Output Name:

Archive Policy

☒ Use the archive policy for the distribution folder(s)

☐ Delete objects older than days hours

☐ Delete objects on (mm/dd/yyyy) at (hh:mm:ss)

☐ Archive items before deletion (only applies if not using archive policy for distribution folder)

Notify Channels

☐ News

[Submit Request](#)

Copyright © 1997-1999 Actuate Corporation

The requester template, REQUEST.ACHTML, can list parameters by group or by name.

Use list parameters to list the parameters alphabetically. Use list paramgroups to list the parameters by group. List parameters can be nested. List paramgroups cannot be nested.

List parmgroups and list parameters are available from the following templates:

- Standard requester page, named REQUEST.ACHTML
- Request confirmation page, named CONFIRM.ACHTML
- Request details page, named REQUESTDETAILS.ACHTML
- Request status page, named STATUS.ACHTML
- Cancelled request confirmation page, named CANCEL.ACHTML

List values command

Generates a request confirmation page.

Syntax

list values [setonly]

Description

Lists the values that the user set for request parameters. For more information about request parameters, see “List parmgroups, list parameters commands,” earlier in this chapter.

The setonly option lists only those parameters for which the user entered a value. You can use this with the list parmgroups command to group the values by parameter groups.

List values is available from the following templates:

- Request confirmation page, named CONFIRM.ACHTML
- Request details page, named REQUESTDETAIL.ACHTML
- Request status page, named STATUS.ACHTML
- Cancelled request confirmation page, named CANCEL.ACHTML

List values can be nested inside the list parmgroups command.

Example

The following example iterates through a values list and displays the values on a request confirmation page. This example is taken from the CONFIRM.ACHTML template:

```

<!-- #Actuate list values -->
<!-- #Actuate content -->
<table>

    <!-- Display the values heading -->

        <tr>
            <td>&nbsp;</td>
            <td><font size="2" face="Arial">Description</font></td>
            <td>&nbsp;</td>

            <!-- If the value is null, display a space -->
            <!-- #Actuate if Value = "" then -->
            <td><font size="2" face="Arial">&nbsp;</font></td>

            <!-- Otherwise, display the value -->
            <!-- #Actuate else -->
            <td><font size="2" face="Arial">$(Value)</font></td>

            <!-- #Actuate end if -->
        </tr>
    </table>

    <!-- #Actuate end list -->

```

List status command

Generates a request status page.

Syntax

list status

Description

The request status page lists the status items available for the current request. List status is available from the following templates:

- Confirmation page, named CONFIRM.ACHTML
- Request status page, named STATUS.ACHTML
- Cancelled request confirmation page, named CANCEL.ACHTML
- Request details page, named REQUESTDETAIL.ACHTML

Example

The following example iterates through a list of status items for reports on a particular e.Reporting Server and displays them on a status page. This example is taken from the STATUS.ACHTML template:

```
The following is the status of running $(ReportName) on $(Server):
<!-- #Actuate list status -->
<!-- #Actuate before -->
<center>
<table border=0 bgcolor=black>
<thead valign=top>
<tr bgcolor="#804080">
<th>Time<th>Status
<tbody>
<!-- #Actuate content -->
<tr bgcolor="E0E0E0"><td>$(DateAndTime)<td>$(StatusText)</tr>
<!-- #Actuate after -->
</table>
</center>
<!-- #Actuate end list -->
```

List members command

Lists every member on a e.Reporting Server.

Syntax

```
list members { (users | roles | groups [...])}
```

Description

The term member refers to any user, role, or group on an e.Reporting Server. You can use list members to generate a pull-down list of users to notify when a particular request completes. The default is to list all members or specify to include only certain types of members, for example, list only users, or roles, or groups. The list is sorted by name.

List members is available from all templates except for the admin page, called ADMIN.ACHTML. It cannot be nested inside another list members command or inside the list grants or list notifications commands.

Example

The following example iterates through a list of the roles that are members on the current e.Reporting Server, and sets up an input checkbox. This example is based on the CHANNELDETAIL.ACHTML template:

```
<!-- #Actuate list members (roles) -->
<!-- #Actuate if MemberName <> "Administrator" and MemberName <>
"All" then -->
    <input type="checkbox" name="__Roles" value="$(MemberName)">
        $(MemberName)
    <!-- #Actuate end if --> <!-- MemberName -->
<!-- #Actuate end list --> <!-- members -->
```

List grants command

Lists all members that are explicitly granted permissions on the current item, either a Report Encyclopedia object or a channel.

Syntax

```
list grants [all] {(users | roles | groups [...])}
```

Description

List grants lists those privileges on an object that are explicitly granted to the current user or role. It does not include those privileges that the current user or role inherited from a role to which they belong. This means that the set of users returned by list grants list does not include the set of users that inherit all their privileges on the object from their roles.

List grants is available from the following templates:

- File detail page, named FILEDETAIL.ACHTML
- Channel detail page, named CHANNELDETAIL.ACHTML
- Request page, named REQUEST.ACHTML

List grants can be nested inside the list versions or list items commands or at the top level in a template for a folder, encyclopedia item, or channel. This command is useful if you want to allow or disallow the current member to execute a report depending upon whether they have privileges on an item.

Use the All option to list all members, not only those that have privileges on the current folder, file, or channel. If the current user does not have

administrator privileges, the list of privileges returned applies only to the current user and the roles to which that user belongs.

For more information about members, see “List members command,” earlier in this chapter.

Example

The following example lists the roles that are granted privileges on the current item. The example is based on the REQUEST.ACHTML template:

```
<!-- #Actuate list grants (roles) -->
<!-- #Actuate if MemberName <> "Administrator" and MemberName <> "All"
then -->
<!-- #Actuate Set RoleFound = "True" -->
<!-- #Actuate end if -->
<!-- #Actuate end list -->
```

List privileges command

Lists all the privileges granted to the current user or role for the current encyclopedia item or channel.

Syntax

list privileges

Description

Lists all the privileges available on the current e.Reporting Server. Use list privileges within a list grants command to list each of the privileges granted to the current user or role for the current Report Encyclopedia item or channel. If you use list privileges in the context of a Report Encyclopedia item, the returned list does not include privileges inherited from a role to which the current user or role belongs.

List privileges is available from the following templates:

- The channel details page, named CHANNELDETAIL.ACHTML
- The file details page, named FILEDETAIL.ACHTML

List notifications command

Creates a list of users and groups to notify when the current request is complete. The list is sorted by name.

Syntax

list notifications

Description

Supports user notification when a request to run a report completes. The standard requester form, REQUEST.ACHTML, displays the list of available channels. To notify a particular channel when a request completes, users click the appropriate check box. You can also modify the STATUS.ACHTML template to show the list of channels to be notified when the user checks the status of the request.

The list notifications command is available from the following templates:

- The confirmation page, named CONFIRM.ACHTML
- The cancellation page, named CANCEL.ACHTML
- The request details page, named REQUESTDETAIL.ACHTML
- The status page, named STATUS.ACHTML

List schedules command

Lists the schedule associated with the current request.

Syntax

list schedules

Description

Supports specifying simple schedules for running reports, such as daily, right now, or once at a specified time and date. The e.Reporting Server's Administrator Desktop also allows you to specify complex schedules, such as twice weekly plus once a month on a specific day. You can view complex

schedules using the list schedules command, even though you cannot specify those commands using ReportCast.

The list schedules command is available from the request details page, named REQUESTDETAILS.ACHTML.

List printerprops command

Lists property names and values for the printer being used for the current request.

Syntax

list printerprops

Description

Provides information about the printer being used to print out the current request. List printerprops is available from the request details page, named REQUESTDETAIL.ACHTML, if the request was either Print or Execute and Print.

List channels command

Lists channels to which the current user is subscribed.

Syntax

list [all] channels

Description

Provides a list of channels to which the user is subscribed. For example, the standard template for the Personal Channel, PERSONALCHCONTENTS.ACHTML, displays the list of channels to which the user is subscribed in the left sidebar. Use the All option to get a list of all available channels on the current e.Reporting Server.

List channels is available from all templates except for ADMIN.ACHTML.

Example

The following example lists all channels available on the current e.Reporting Server:

```
<!-- #Actuate list all channels -->
```

List reportservers command

Lists the report servers to which ReportCast has open connections.

Syntax

list reportservers

Description

ReportCast opens a connection to a particular e.Reporting Server each time a user accesses an e.Reporting Server through ReportCast. Users with administrative privileges can check the open connections from ReportCast to a particular e.Reporting Server from the ReportCast administration page. Open connections use system resources. If response time is slower than desirable, you can close open connections from the administration page.

List reportservers is available from the admin page, named ADMIN.ACHTML.

List pathlinks command

Syntax

list pathlinks

Description

Creates a list of the path links that apply to the current folder, file, or .ROD. The list is empty if there is no current folder, file, or .ROD. Use this command to find the individual elements of a folder, file, or .ROD's complete path name. For example, list pathlinks allows cascading style sheet properties to control the colors of Report Encyclopedia objects' links on the reporting web page.

List pathlinks is available from the following templates:

- The folder list page, called FOLDERLIST.ACHTML
- The file details page, called FILEDETAIL.ACHTML
- The confirmation page, called CONFIRM.ACHTML
- The request details page, called REQUESTDETAIL.ACHTML
- The status page, called STATUS.ACHTML
- The cancellation page, called CANCEL.ACHTML

Example

The following example is a fragment from FOLDERLIST.ACHTML. It displays the folder path name with each directory name separated by a small folder icon:

```
<!-- #Actuate list pathlinks -->
<!-- #Actuate ifempty -->
<tr>
  <td valign="bottom" nowrap>
    <table border="0" cellpadding="0" cellspacing="0">
      <tr>
        <td></td>
        <td><font size="3" face="Arial"><strong>$(ServerAlias)</strong>
</font></td>
      </tr>
    </table>
  </td>
</tr>
<!-- #Actuate before -->
<tr>
  <td valign="bottom" nowrap>
    <table border="0" cellpadding="0" cellspacing="0">
      <tr>
        <td></td>
        <td><font size="3" face="Arial"><strong>
<!-- #Actuate content -->
<!-- #Actuate if Hyperlink <> "" then -->
          <a href="$(Hyperlink)">$(DisplayName)</a>&nbsp;
&nbsp;
<!-- #Actuate else -->
          $(DisplayName)
<!-- #Actuate end if -->
<!-- #Actuate after -->
        </strong></font></td>
      </tr>
    </table>
  </td>
</tr>
```

```
</table>
</td>
</tr>
<!-- #Actuate end list -->
```

List expirationpolicies command

Syntax

list expirationpolicies

Description

Creates a list of the object aging expiration policies that apply to the current Report Encyclopedia file. The expirationpolicies command is available from the file detail page, FILEDETAIL.ACHTML.

List tocitems command

Syntax

list tocitems

Description

Lists the items in the DHTML report's Table of Contents. The list tocitems command is available from the view table of contents page, named VIEWTOC.ACHTML.

List criteria command

Syntax

list criteria

Description

Lists the search criteria for a search query. The list criteria command is available from the search results page, named SEARCHRESULTS.ACHTML.

List resulthits command

Syntax

list resulthits

Description

Lists the search result hits returned by the search query. The list resulthits command is available from the search results page, named SEARCHRESULTS.ACHTML.

List resultfields command

Syntax

list resultfields

Description

Lists the fields that are part of the search query's results.

You can call list resultfields within the scope of a particular search result hit. Doing so allows you to access the value and component ID for each field in that result hit.

You can call list resultfields outside the scope of any particular search result hit. Doing so allows you to access only the name of each field contained in the search query's results.

The list resultfields command is available from search results page, named SEARCHRESULTS.ACHTML.

List viewpageformats

Syntax

list viewpageformats

Description

Lists the available formats for displaying DHTML report pages. Use the \$(FormatValue) variable to iterate through the list of available formats.

You can use the viewpageformats values to create or modify the ?ViewPage or ?GetReportData URL dynamically in ReportCast templates.

The list of available formats depends upon the configuration of Actuate e.Reporting Server. The number of installed products, add-ins, and custom converters determines the formats that are available.

The following are the supported display formats.

- XMLDisplay
- XMLCompressedDisplay
- DHTML
- DHTMLLong
- DHTMLRaw
- PDF

The list viewpageformats scripting language command is available to the following pages:

- The search request page, named SEARCHREQUEST.ACHTML
- The search results page, named SEARCHRESULTS.ACHTML
- The navigation bar, named VIEWNAV.ACHTML
- The report table of contents, named VIEWTOC.ACHTML

Example

The following example lists the available search formats.

```
<P>Here are the available search formats:<P>
<!-- #Actuate list searchformats -->
<!-- #Actuate before -->
<!-- #Actuate content -->
```

```
$(FormatValue)<br>
<!-- #Actuate after -->
<!-- #Actuate end list -->
```

List searchformats

Syntax

list searchformats

Description

Lists the supported formats for presenting the results of search operations. Use the \$(FormatValue) variable to iterate through the list of available formats.

You can use the values searchformats values to create or modify the ?searchReport URL.

The list of available formats depends upon the configuration of Actuate e.Reporting Server. The number of installed products, add-ins, and custom converters determines the formats that are available. For example, the ANALYSIS format is available if the e.Analysis option is installed.

The following are the supported search formats.

- XMLDisplay (DISPLAY)
- Download formats:
 - CSV
 - TSV
 - ANALYSIS

The list searchformats scripting language command is available to the following templates:

- The search request page, named SEARCHREQUEST.ACHTML
- The search results page, named SEARCHRESULTS.ACHTML
- The navigation bar, named VIEWNAV.ACHTML
- The report table of contents, named VIEWTOC.ACHTML

List searchformats example

The following example lists the available search formats.

```
<P>Here are the available search formats:<P>
<!-- #Actuate list searchformats -->
<!-- #Actuate before -->
<!-- #Actuate content -->
$(FormatValue)<br>
<!-- #Actuate after -->
<!-- #Actuate end list -->
<P>
```

If command

Conditionally includes HTML in the generated web page.

Syntax

```
if expr then
  elseif expr then
  else
  end if
```

expr

An expression.

Expressions have the following formats:

```
expr and expr
expr or expr
(expr)
not expr
var = "string"
var <> "string"
var = var
var <> var
```

Description

Allows template developers to include HTML in their generated web pages based on specific conditions. For example, use the If command to provide a particular format for particular types of files, or hide files the current user is not authorized to see.

Include command

Includes another template file with the current template file.

Syntax

```
include filename
```

Description

Including other template files with the current template file provides reusability for your HTML template code. The included template file can include other template files, which may include other template files, and so on. Included template files must be both readable and located in the current search path's directory tree. In other words, they must be local to the report server. You cannot include templates that reside on other file systems or directory structures.

The Include command is useful for including a standard element in several templates. It simplifies maintenance to make changes to one file, rather than to several files. For example, include the standard company logo in the LOGO.ACHTML template file and standard company copyright information and links to help in the COPYRIGHT.ACHTML template.

Example

The following example includes a template in the current template file:

```
<!-- #Actuate include "stdfooter.achtml" -->
```

With command

Makes the specified object's variables and list statements available.

Syntax

```
with objectname  
  statements  
end with
```

Description

Includes a named object's variables and list statements in the current object's scope. Objects have the following values.

Object value	Description
ReportDefinition	Makes available the Report Definition for the current request, which provides the parameter, parameter group lists, and other properties.
InputObject	Makes available the Folder Item for the .ROX file of the request.
OutputObject	Makes available the Folder Item for the .ROI or .ROW of the request.
ValuesObject	Makes available the Folder Item for the .ROV file of the request.

Example

The following example makes the ReportDefinition object available, then iterates through its parameter groups list:

```
<!-- #Actuate with ReportDefinition -->
<!-- #Actuate list paramgroups -->
<!-- #Actuate content -->
<!-- (Actions taken on parameter groups here) -->
<!-- #Actuate end list -->
<!-- #Actuate end with -->
```

Set command

Assigns a value to a variable.

Syntax

```
set var="string"
```

```
set var1=var2
```

var, var1, var2

Unique variable names. Variable names are case-insensitive.

string

The value of the character string to assign to var.

Description

Assigned values can be character strings or other variables' values. Once a variable is assigned, it is available within the current template using the following syntax:

`$(VariableName).`

Example

The following example checks whether or not a folder contains items, and if so, sets the variable FolderHasItems to True. The following example is taken from the FOLDERLIST.ACHTML template:

```
<!-- #Actuate list items -->
<!-- #Actuate content -->
<!-- #Actuate if FolderPath<>"/" or ( ItemBaseName<>"$$$TempROVs" and
ItemBaseName<>"$$$CUTCOPY" ) then -->
<!-- #Actuate Set FolderHasItems = "True" -->
<!-- #Actuate end if -->
<!-- #Actuate end list -->
```

ReportCast scripting language variables

The ReportCast scripting language variables are placeholders for data within the template files. ReportCast substitutes actual data values for the variables when it generates the web page. The variable substitution process enables creating HTML pages dynamically, depending on the directive sent. Dynamic HTML pages support page generation that reflects the constant data changes on the e.Reporting Server.

ReportCast variables provide current, run-time access to information about a wide variety of topics including the ReportCast services, e.Reporting Server, the available folders and files, schedules, ReportCast channels, requests, privileges, and parameters.

Variables use the following syntax:

`$(variable)`

All variable names are case-insensitive. For example, the following command tells ReportCast to generate a page that displays the name of a report and its owner's name:

The name of the report is `$(ReportName)` and the owner is `$(OwnerName)`.

If the report name is SalesEastQ497.roi, and the owner is SalesMgr, the generated page displays:

The name of the report is SalesEastQ497 and the owner is SalesMgr.

ReportCast variables

ReportCast variables return information about the machine running ReportCast, as well as information about ReportCast itself. ReportCast variables are available on all templates. The following tables list the scripting language variables.

Variable	Purpose
Today	Displays the current date on the machine running ReportCast. Date format is localized.
Now	Displays the current time on the machine running ReportCast. Time format is localized.
AgentVersion	Displays the product name and version of ReportCast.
Host	Displays the name of the web server machine that is connected to the ReportCast server. This is the web server machine itself, not a proxy server. To resolve Host to a host name, DNS on the ReportCast Server machine must be able to find a host name for the address of the web server machine. Otherwise, this value is blank.
TemplateLoadDateAndTime	Displays the date and time that ReportCast's template file cache was last cleared.
StartDateAndTime	Displays the date and time that ReportCast was started.
ElapsedTime	Displays the elapsed time since ReportCast was started.
NumberOfCommands	Displays the number of commands that ReportCast has processed since the last time it was started.
AdminURL	Displays the URL to ReportCast's administration page.
ReportServerName	Displays the report server name that the user specified during login. ReportServerName is available only from the login form, and only if ReportCast is not set up with Basic Authentication.

Web server variable

The following table lists the ReportCast variable that returns information about the web server on which ReportCast is running. This variable is available on all templates.

Variable	Purpose
WebServerVersion	Displays the product name and version of the web server in which ReportCast is running.

Web request variables

The following table lists the ReportCast variables that return information about the current command that ReportCast is processing. They are available on all templates.

Variable	Purpose
CommandName	Displays the name of the command appended to the URL.
URL	Displays the full URL sent to ReportCast.
ObjectURL	Displays the URL only for the object that is the target of the URL.
UserName	Displays the name of the current user.
UserAgent	Displays the contents of the User Agent, for example, Mozilla/3.01 WinNT.
RemoteAddress	Displays the dotted-decimal IP address of the browser.
RemoteHost	Displays the fully resolved host name of the browser.
TemplateName	Displays the name of the template without the directory path.
PreviousURL	Displays the URL of the web page previous to the current page.

e.Reporting Server variables

The following table lists ReportCast variables that return information about the e.Reporting Server named in the current request's URL. These variables are available in all templates except the ReportCast administration page.

Variable	Purpose
Server	Displays the name of the e.Reporting Server.
ServerVersion	Displays the full product name and version of the e.Reporting Server.
ScheduledRequestsURL	Displays the URL for the list of scheduled requests on the e.Reporting Server.
ActiveRequestsURL	Displays the URL for the list of active requests on the e.Reporting Server.
CompletedRequestsURL	Displays the URL for the list of completed requests on the e.Reporting Server.
HomeFolderURL	Displays the URL for the listing of the current user's home folder on the e.Reporting Server. If no home folder is defined for the current user, HomeFolderURL has the same value as RootFolderURL.
RootFolderURL	Displays the URL to the listing of the root folder of the e.Reporting Server.
ChannelsURL	Displays the URL to the listing of the current user's channels.
ServerAlias	Displays the descriptive string name of the e.Reporting Server. If there is no descriptive name, ServerAlias is the e.Reporting Server's machine name.
IsAdministrator	True if the current user belongs to the administrator group of the current e.Reporting Server. False otherwise.
ServerFeatureLevel	Indicates the e.Reporting Server level. 0 is None, 1 is Standard, and 2 is Advanced.

Variable	Purpose
ViewFormatPreference	Contains the current user's format preference for viewing reports. DHTML, ROI, or SYSTEM DEFAULT. SYSTEM DEFAULT uses the viewing preference set by the e.Reporting Server administrator.
ViewFormatDefault	Displays the system-wide default set by the e.Reporting Server administrator, either DHTML or ROI.

Note that the value for ServerAlias comes from the ACSERVERALIAS property of the root folder on the e.Reporting Server. Use the Navigator to set the e.Reporting Server's name property, which sets the ACSERVERALIAS value.

View session variables

The following table lists the ReportCast variables that return information about the current view session. The view session variables are available whenever the user views a DHTML report.

Variable	Purpose
CurrentPage	The number of the DHTML report page currently being viewed.
CurrentTOCParentID	The component ID of the current Table of Contents entry, or node.
TotalPages	The total number of pages in the DHTML report.
FormatValue	The value of a supported search or viewing format.

Table of contents item variables

The following table lists the ReportCast variables that return information about the current table of contents item. The table of contents item variables are available whenever the user makes use of a DHTML table of contents.

Variable	Purpose
ComponentID	The current table of contents item's component ID.
Expandable	TRUE if the table of contents item has subitems that appear when a user expands the item by clicking on the item's "+" sign, FALSE otherwise.
Text	The displayable name of the current table of contents item.

Search criterion variables

The following table lists the ReportCast variables that return information about the current search criterion. The search criterion variables are available whenever the user performs a search on a DHTML report.

Variable	Purpose
ClassName	The name of the class to which the current search criterion belongs.
ControlName	The name of the control that the current search criterion applies to. For example, "Office name."
ComponentID	The component ID of the control that the current search criterion applies to.
SearchValue	The expression that the control's value must match. For example, "Boston," or ">1000." SearchValue is blank if the user did not enter a search expression for the control, but selected it as part of the search results.
Selected	TRUE if the user selected the control as part of the search results, FALSE otherwise.

Search result set variables

The following table lists the ReportCast variables that return information about the current search result set. The search result set variables are available whenever the user examines the results returned by a search on a DHTML report.

Variable	Purpose
TotalNumberOfHits	The total number of result hits returned by the search.
NumberOfHitsOnPage	The number of hits displayed on the search results page.
StartHitNumber	The index of the first hit displayed on the search results page. The range is 1–TotalNumberOfHits.
EndHitNumber	The index of the last hit displayed on the search results page. The range is 1–TotalNumberOfHits.
Format	The value of the format URL.
Frameset	The value of the frameset URL.

Search result hit variables

The following table lists the ReportCast variables that return information about the current search result hit. The search result hit variables are available whenever the user examines a specific search result hit.

Variable	Purpose
ComponentID	The current search result hit's component ID.

Search result field variables

The following table lists the ReportCast variables that return information about the current search result field. The search result field variables are available whenever the user examines a specific search result field.

Variable	Purpose
Value	The displayable value of the current search result field.

Current folder variables

The following table lists the ReportCast variables that return information about the current folder. The current folder variables are available whenever the user displays a page that describes a folder, for example, the folder listing page FOLDERLIST.ACHTML.

Variable	Purpose
FolderPath	Displays the full path name of the current folder.
FolderName	Displays the name of the current folder.
ParentFolderPath	Displays the full path name of the folder that contains the current folder.
ParentURL	Displays the URL of the parent folder. This variable's value is empty ("") if there is no parent folder.
FolderHeader	Displays the description of the folder's contents.
FolderTitle	Displays the title to be used in the HTML <TITLE> tag.
LinkedFolderPath	Displays a series of lines that provide a hierarchical path to the current folder.
LastModified	Displays the date and time that the folder was last modified.

Folder item variables

The following table lists the ReportCast variables that return information about folder items. Folder items are both folders and the files inside a folder. These variables are available for each item in a list items, list files, and list folders list, and from the File detail page.

Variable	Purpose
LastModified	Displays the date and time that the item was last modified.
LastModifiedBy	Displays the name of the user who last modified the item.
PathName	Displays the full path name of the item on the e.Reporting Server.
ItemName	Displays the name of the current item.
ItemBaseName	Displays the item's base name, without the file name extension.

Variable	Purpose
ItemPathName	Displays the full path name of the item.
Created	Displays the item creation date and time.
OwnerName	Displays the name of the item's owner.
URL	Displays the URL of the item.
ObjectID	Displays the item's unique object ID.
FileType	Displays the item's file type.
AbbreviatedType	Displays the extension of the file, for example, .ROX, .ROI.
FileSize	Displays the size of the item.
AbbreviatedFileSize	Displays the size of the item, abbreviated according to the localized display format. For example, 12.5MB, 923KB.
FileVersion	Displays the version number of the item.
FileVersionName	Displays the descriptive version name of the file item. If there is no descriptive name, ReportCast displays Version <n>, <n> is the version number.
IsBundledOutput	TRUE if the object is a .ROW file or a third-party report consisting of HTML pages, FALSE otherwise. View bundled files using the ?View or ?ViewDefault directives.
IsExecutable	TRUE for .ROX, .ROV, and third-party report executable files, FALSE otherwise. You can request these files using the ?Request directive.
LatestURL	The URL to the most recent version of the file.
LinkedFolderPath	A series of lines that provide a hierarchical path to the current folder item's properties.
NewItem	True if the current item is the first version of a file to be listed in the list items or list files commands, False otherwise. The first version is not necessarily Version 1. It is simply the first in this particular list. Its position depends on the sorting options used.
NumVersions	Displays the number of versions of the file that correspond to the current item.
Comment	Displays a lengthy description of the file, stored as a property on the e.Reporting Server. Displays on detail pages in the Navigator. Comments can be of any length.

Current request folder

The following table lists the ReportCast variables that return information about the request folder. These variables are available from the Request folder page or the Channel contents page.

Variable	Purpose
FolderName	Displays the name of the request folder: Scheduled, Active, or Completed.
Entries	Displays the number of requests in the request folder.

Report definition

The following table lists the ReportCast variables that return information about a report. These variables are available from the Standard Requester page and the Request confirmation page.

Variable	Purpose
ReportName	Displays the name of the report executable, without the file extension.
ReportPath	Displays the complete name of the report executable.
ReportOutputPath	Displays the default location of the output file for the report (path and file name).
LastModified	Displays the date and time the report executable was last modified.
DefaultDate	Provides a default date value (the current date) for the report requester form, in the correct localized format.
DefaultTime	Provides a default time value (the current time) for the report requester form, in the correct localized format.
ReportURL	Displays the URL of the report executable.
CancelURL	Displays the URL for cancelling the current request. Available only from the Request Confirmation page.
ViewURL	Displays the URL for viewing the output of the request. Available only from the Request Confirmation page.

Variable	Purpose
DownloadURL	Displays the URL for downloading the output of the current request. Available only from the Request Confirmation page.
StatusURL	Displays the URL for the status page of the current request. Available only from the Request Confirmation page.
RequestURL	Displays the base URL containing the ID of the current request, without the query portion. RequestURL can be combined with queries in other templates to produce URLs other than the request's URL.

Request item variables

The following table lists the ReportCast variables that return information about items in a list requests list. These variables are available from the Request status and Request detail pages.

Variable	Purpose
RequestID	Displays the unique ID of the current request.
RequestType	Displays the type of the current report-generation request: Print or Execute.
RequestedBy	Displays the name of the person submitting the request.
RequestStatus	Displays the status of the current request: Scheduled, Active, Completed, or Failed.
OutputName	Displays the base name of the report document that will be generated from the current request.
OutputLocation	Displays the name of the folder containing the request's output file.
OutputPathName	Displays the complete name of the output file, including the full path name and file extension.
ReportName	Displays the name of the requested report executable (.ROX).
ReportPathName	Displays the complete name of the requested report's executable, including the full path name and file extension.
Priority	Displays the numeric priority of the request.

Variable	Purpose
StartTime	Displays the date and time at which the current request started to run.
CompletedTime	Displays the date and time at which the current request completed.
Headline	Displays the request headline. This variable is valid only for successfully completed requests.
ReportURL	Displays the URL of the report executable.
CancelURL	Displays the URL of the command to cancel the request.
ViewURL	Displays the URL for viewing the output of the request. This variable is valid only for successfully completed requests.
DeleteURL	Displays the URL for the command to delete status information for the current request.
StatusURL	Displays the URL of the status page for the current request.
RequestURL	Displays the base URL for the ID of the current request, without the query portion.
DetailURL	Displays the URL for the request's detail page.
ReportVersion	Displays the version number of the requested report executable.
ReportVersionName	Displays the version name of the requested report executable.
CurrentSize	Displays the current size of the output of the request, in bytes.
AbbreviatedCurrentSize	Displays the current size of the output of the request, in KB, abbreviated.
CurrentPages	Displays the current number of pages of the request's output.
ElapsedTime	Displays the time expired since the start of the request.
ProcessGroup	Displays the name of the process group running the request.
OutputVersion	Displays the version number of the completed request's output file.
OutputVersionName	Displays the version name of the output file of the completed request.

Variable	Purpose
Size	Displays the size of the completed request's output file.
AbbreviatedSize	Displays the size of the completed request's output file, in KB, abbreviated.
Pages	Displays the number of pages in the completed request's output file.
NextStartTime	Displays the time that the request is scheduled to start.
ValuesFileName	Displays the name of the scheduled request's .ROV file.
PrintRange	Displays the range of pages to be printed for the scheduled request.

Member variables

The following table lists the ReportCast variables that return information about members including users, groups, and roles. Member lists include the list of all members on an e.Reporting Server, the list of members that have privileges for a folder item or channel, and the list of members to be notified about a request.

Variable	Purpose
MemberName	Displays the name of the current user, group, or role.
MemberType	Displays the member's type: User, Group, or Role.

Privilege variables

The following table lists the ReportCast variables that return information about privileges for a current folder item or channel and a current member. These variables are available from the list privileges command.

Variable	Purpose
PrivilegeName	Displays the name of the current privilege.
Granted	Displays True or False depending on the current privilege, folder item or channel, and member.

Request status variables

The following table lists the ReportCast variables that return information about the current request's status. These variables are available from the List status command.

Variable	Purpose
DateAndTime	Displays the date and time of the status element.
StatusText	Displays the text of the status element.

Parameter group variable

The following table lists the ReportCast variable that returns information about related groups of report parameters. These variables are available from the List paramgroups command.

Variable	Purpose
GroupName	Displays the parameter group's name.

Parameter variables

The following table lists the ReportCast variables that return information about report parameters. These variables are part of report definitions and parameter groups. These variables are available from the list parameters command. Use the List parameters command either from the top level of a report definition or within a List paramgroups command.

Variable	Purpose
Default	Displays the parameter's default value.
Value	Displays the value that the user entered for the current parameter.
Type	Displays the type of information that the parameter should contain, for example, String, Number.
Alias	Displays the parameter's alias, if one exists, or "" if there is no alias.
Name	Displays the parameter's actual name. That name is what ReportCast passes to the factory.
Description	Displays a description of the parameter's name. This description is the same as Alias if Alias is not empty.

Variable	Purpose
Required	Displays True if this parameter is required. Displays False otherwise.
AdHoc	Displays True if this is an ad hoc parameter. Displays False otherwise.
Hidden	Displays True if this is a hidden parameter. Displays False otherwise.
HideText	Displays True if this parameter's value must be masked. Displays False otherwise. Useful for masking values such as passwords in the authentication dialog box.

Schedule variable

The following table lists the ReportCast variables that return information about the current request's schedule. These variables are available from the List schedules command.

Variable	Purpose
ScheduleDescription	Displays the text description of the request's schedule.

Printer variables

The following table lists the ReportCast variables that return information about the printer being used for the current request. These variables are available from the list Printerprops command.

Variable	Purpose
Name	Displays the name of the current printer property.
Value	Displays the value of the current printer property.

ReportCast Channel variables

The following table lists the ReportCast variables that return information about the current channel. These variables are available from the list channels command, or from the Channel detail or Channel contents pages.

Variable	Purpose
ChannelName	Displays the name of the current channel.
ChannelURL	Displays the URL of the current channel.
Subscribed	Displays True if the current user is subscribed to the current channel. Displays False otherwise.
Expiration	Displays the number of days an item remains on the current channel before it is removed.
PollingInterval	Displays the number of minutes between browser refreshes of the current channel's contents. This number is formatted appropriately for the current configuration. For example, commas separate every three digits in English: 1,000.
RawPollingInterval	Displays the number of seconds between browser refreshes of the current channel's contents. This number is not formatted.
NextPollTime	The time at which the browser must refresh the contents of a channel. This value is an HTTP date, and can be used in a <meta> HTML tag to cause a page to expire from the browser's persistent cache. Current information about HTTP dates is available from the HTTP specification available from the World Wide Web Consortium's web site, http://www.w3.org/ .
SmallIconURL	Displays the URL of the small version of the custom icon for the current channel. This value is empty ("") if there is no custom icon.
LargeIconURL	Displays the URL of the large version of the custom icon for the current channel. This value is empty ("") if there is no custom icon.
Comment	Displays a description of the channel, with HTML tags. The description may be of any length.
PlainComment	Displays a description of the channel, without the HTML tags. Use PlainComment in multiline input tags on an HTML form.

Error variables

The following table lists the ReportCast variables that return information about error conditions that occur on e.Reporting Server requests. Error variables are available from the Error page.

Variable	Purpose
ErrorClass	Displays the type of error that occurred: 1 e.Reporting Server Connection Error 2 e.Reporting Server Message Error 3 ReportCast Internal Error 4 ReportCast Parser Error
ErrorText	Displays a description of the specific error condition, if possible. The description is localized.
RawErrorText	Displays a description of the specific error condition. The text is not localized.
ErrorNumber	Displays the numeric code for the error. This is an HTTP status code for all but ReportCast internal errors and parser errors.
Server	Displays the name of the report server on which the error occurred.

Internal ReportCast security

This chapter contains the following topics:

- About ReportCast security
- Providing security for ReportCast administration
- Logging on and authenticating users
- Security and the Apache web server
- Customizing the Login dialog box
- Setting the idle user session timeout
- Modifying the HTTP unauthorized status code
- About the ReportCast Security Extension
- Development considerations
- Enabling RCASE and RCSE

About ReportCast security

A reporting web site is accessible to any user who has a web browser and the URL to the site. How do you ensure that users access only those Report Encyclopedia objects for which they have permission? How do you protect sensitive reports, or ensure that only authorized users can perform administrative tasks for ReportCast? This chapter discusses the ReportCast security features and how to use them to provide secure access to your web reporting site.

The types of security you can provide for ReportCast are:

- **Administrative security.** Assign an administrator login and password to provide security for performing ReportCast's administrative tasks. This security applies to tasks specific to ReportCast only. They are not related to e.Reporting Server administration.
- **Default user authentication.** Use the default ReportCast and e.Reporting Server facilities to ensure that users access only those reports and other Report Encyclopedia objects for which they have permission.
- **User authentication using ReportCast Security Extension (RCSE) or ReportCast Agent Security Extension (RCASE).** Use the RCSE or RCASE to customize and control the user login and authentication process.

You can also customize the Login dialog box and specify the unauthorized status code that ReportCast returns when authentication fails.

Actuate fully supports HTTPS (Secure Socket Layers, or SSL). The default port for HTTPS is 443. For information about configuring the port numbers, see "Configuring ReportCast port numbers" in Chapter 1, "About Actuate ReportCast."

Setting the ReportCast security variables

Enabling the ReportCast Agent Security Extension and ReportCast Server Security Extension and setting up secure facilities involves using registry keys on NT or environment variables on UNIX to set values for the RCSE security variables.

How to set security variables on NT

Create new registry keys using Regedit. Be sure to back up your system before running Regedit:

- 1 From the Start menu, Run Regedit.
The Registry Editor window displays.

- 2 Choose HKEY_LOCAL_MACHINE\Software\Actuate\Actuate ReportCast Server\5.
- 3 Select Actuate ReportCast.
- 4 Choose Edit.
The Edit pull-down menu displays.
- 5 Choose New→String Value.
A new value item displays in the right side of the registry.
- 6 Type the name of the security variable.
For example, if you are setting the registry key for the AC_REPORTCAST_ADMINISTRATOR_USER_NAME security variable, type:
AC_REPORTCAST_ADMINISTRATOR_USER_NAME
- 7 Set the security variable:
 - 1 Select the security variable name.
 - 2 Right-click on the security variable name, then choose Modify.
The Edit String dialog box appears.
 - 3 Type the value for the security variable in the Value data: field.
 - 4 Choose OK.

How to set security variables on UNIX

Set the security environment variables in the user environment in which the ReportCast runs. In other words, you must be logged on as the user that runs ReportCast. That is usually the same user that runs the web server, unless ReportCast runs separately from the web server as on HP-UX systems.

The easiest method is to set the security variables and export them in the ReportCast startup script, startreportcast.sh. The ReportCast installation places the startreportcast.sh script in the <webserver-home>/actuate//bin directory, where <webserver-home> is the root of your web server installation. Add the setenv and export statements to the startup script just before the script invokes the ReportCast server, reportcastsrvr.

For example, to set and export the AC_REPORTCAST_ADMINISTRATOR_USER_NAME security variable:

```
AC_REPORTCAST_ADMINISTRATOR_USER_NAME=<user>
export AC_REPORTCAST_ADMINISTRATOR_USER_NAME
<user> is the administrator user name.
```

Providing security for ReportCast administration

You perform administrative tasks for ReportCast from the ReportCast administration web page:

`http://<webserver>/acweb/__admin`

<webserver> is the name of the web server that is running ReportCast.

The ReportCast administration web page accesses administrative information for ReportCast. That page cannot be used to administer the e.Reporting Server.

To enable secure access to ReportCast's administration page, use these registry keys for NT, or these environment variables for UNIX:

- `AC_REPORTCAST_ADMINISTRATOR_USER_NAME` sets the administrator's user name
- `AC_REPORTCAST_ADMINISTRATOR_PASSWORD` sets the administrator's password

You can set a user name and password or leave both blank. For example, a ReportCast installation used for testing purposes only or accessible only internally, may not require security. Setting these values for ReportCast does not affect access to your Report Encyclopedias. For information about setting these security variables, see "Setting the ReportCast security variables," earlier in this chapter.

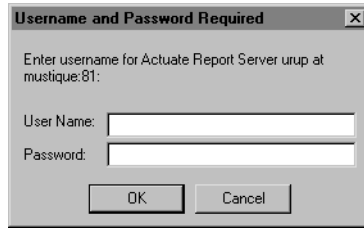
Logging on and authenticating users

Users can log on using their e.Reporting Server user names and passwords. Another option is using anonymous login. If the anonymous login is set up in the e.Reporting Server, ReportCast automatically logs on users as anonymous.

Logging on as anonymous

The anonymous user allows users to access reports on a particular e.Reporting Server without having to go through a login screen or any sort of authentication. e.Reporting Server administrators set up the anonymous user in the Administrator Desktop, configure it with View Properties privileges on all but the most sensitive objects, and do not assign a password.

When a user attempts to access a secure object, ReportCast displays the Login dialog box.



The user must enter a valid user name and password to be able to access the secure object. ReportCast reconnects the user to the e.Reporting Server with the new authentication information and privileges.

Most browsers put already-viewed pages into their cache. That makes viewing web pages faster, but the pages could be out of date. If you reconnect to the e.Reporting Server as a different user, make sure that the web reporting pages you view are current. If they are not, perform a forced reload of the page:

- In Netscape Navigator, hold down the Shift while clicking Reload.
- In Microsoft Internet Explorer, clear the cache and then reload the page.

For more information about setting up the anonymous user, see Chapter 2, "Managing Report Encyclopedia security," in *Administering the Report Encyclopedia*.

Logging on as an e.Reporting Server user

If the e.Reporting Server administrator did not set up an anonymous user, the Login dialog box appears when the user first accesses the e.Reporting Server. When the user enters a valid user name and password, the initial e.Reporting Server view appears in the browser.

Authenticating users using forms

To log in as a different user using a dialog box, users must first log out, then shut down their web browser, restart their web browser, and log back in. If you enable form-based login authentication, users can log out and log back in as a different user without shutting down their web browser.

Users with administrator privileges can customize the login and logout forms. Customize the login form by modifying LOGIN.ACHTML. Customize the logout form by modifying LOGOUT.ACHTML.

Form-based authentication is the default for Apache web servers. For other web servers, enable form-based login authentication by setting the AC_REPORTCAST_USE_LOGIN_FORM registry key (on NT) or environment variable (on UNIX) to TRUE.

Security and the Apache web server

The Apache web server default security process is slightly different than other Unix web servers' default security processes.

Users log in using a form, instead of logging in using a dialog box. For more information, see "Authenticating users using forms," earlier in this chapter.

Customizing the Login dialog box

You can customize the Login dialog box to a certain extent. The Login dialog text is:

- Enter username for <custom text> at <webserver>:
 - <custom text> is the text that you can customize.
 - <webserver> is the name of the web server to which the user wants to connect. You can include the string \$(Server) in your custom text to display the name of the server the user is logging into.

Customize the Login dialog box using the security variable `AC_REPORTCAST_AUTHENTICATION_PROMPT`. For information about setting this security variable value, see "Setting the ReportCast security variables," earlier in this chapter.

For example, setting the following value for `AC_REPORTCAST_AUTHENTICATION_PROMPT`:

logging in to MyCorp's \$(Server) server

results in the following Login dialog prompt to the Sales web server from the ERegion server:

Enter username for logging in to MyCorp's Sales server at ERegion:

Setting the idle user session timeout

If a user is idle for more than 1800 seconds (30 minutes), they must log in again. Users with administrator privileges can change the idle time value by setting the environment variable `AC_REPORTCAST_LOGIN_TIMEOUT`. Specify the length of time that user sessions can be idle, in seconds. For example, to set the timeout to 15 minutes, add the following line to the startup script just before the script invokes the ReportCast server, `reportcastsrvr`:

```
AC_REPORTCAST_LOGIN_TIMEOUT=900
export AC_REPORTCAST_LOGIN_TIMEOUT
```

Modifying the HTTP unauthorized status code

When user authentication fails, ReportCast sends the HTTP unauthorized status code, 401, to the browser. You can modify the HTTP unauthorized status code that ReportCast sends to the browser by setting the `AC_REPORTCAST_UNAUTHORIZED_STATUS_CODE` variable.

Authentication fails if the ReportCast Agent Security Extension or the ReportCast Server Security Extension returns a non-zero value from an authentication function or login to the e.Reporting Server fails. By default, ReportCast sends back the status code of 401 if authentication fails. The 401 error causes the browser to prompt the user for user name and password. This security variable allows the administrator to prevent clients from trying to authenticate by returning a value other than 401. For information about setting this security variable, see “Setting the ReportCast security variables,” earlier in this chapter.

If you want to customize the text to display if authentication fails, you can customize the `error.achtml` template. For more information about customizing the ReportCast templates, see “About ReportCast templates,” in Chapter 3, “Working with ReportCast templates.”

About the ReportCast Security Extension

The Actuate ReportCast Security Extension controls the way that information associated with requests from the web browser is used by ReportCast to log on to the e.Reporting Server.

The ReportCast Security Extension consists of the following:

- ReportCast Agent Security Extension (RCASE). RCASE resides with the ReportCast Agent, as part of the web server. You can use RCASE to filter and change HTTP headers, including user authentication information.
- ReportCast Server Security Extension (RCSE). RCSE resides with the ReportCast Server. RCSE receives the HTTP headers forwarded from RCASE. RCSE can authenticate the user information with a security database or send the user information to the e.Reporting Server for authentication.

Overview

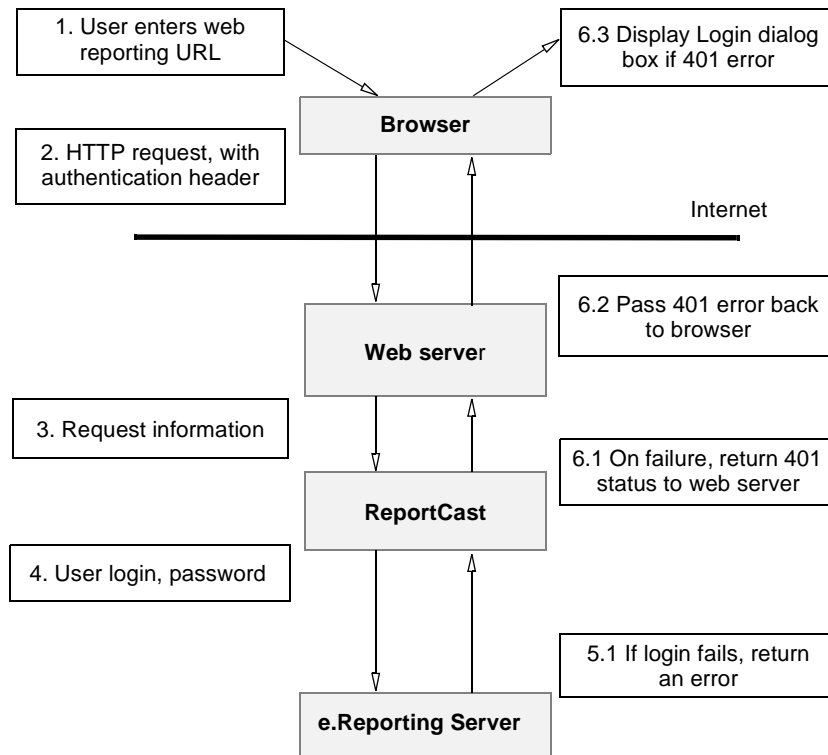
The Actuate ReportCast Security Extension replaces the default user authentication in ReportCast with external routines. Using the security extension enables developers to customize the interaction among the web

browser, ReportCast Agent, ReportCast Server, and e.Reporting Server to control the user login and authentication process.

Authentication means determining whether or not the user is who they say they are. Authorization means determining whether or not the user has access rights to a resource.

You usually integrate RCASE and RCSE with the e.Reporting Server Security Extension, RSSE. For more information about RSSE, see *Integrating Actuate e.Reporting Server*.

The following diagram illustrates the default authentication process.



The following steps describe the default authentication process:

- 1 The user enters the web reporting home page URL in the browser.
- 2 The browser passes the HTTP request to the web server, including the authentication header.
- 3 The web server recognizes the acweb keyword and passes the request information to ReportCast.

- 4 ReportCast maps the Basic Authentication header to the user login and password, then passes the user login and password to the e.Reporting Server. If there is no Basic Authentication header, the user login becomes anonymous and the password is null (""). The anonymous login works only if it is already set up on the e.Reporting Server.
- 5 The e.Reporting Server processes the user login and password:
 - 1 If login fails, the e.Reporting Server returns an error to ReportCast.
 - 2 If login succeeds, the e.Reporting Server returns successful to ReportCast.
- 6 If login failed:
 - 1 ReportCast returns the HTTP 401 status to the web server.
 - 2 The web server returns the HTTP 401 status to the browser.
 - 3 The browser displays the Login dialog box, and the process begins again.

This authentication process is effective most of the time. However, it may require the user to log on multiple times if there is additional security in place on the reporting web site. This authentication process does not support the Microsoft Internet Information Server Windows NT Challenge/Response security feature. It supports only Basic Authentication.

Using ReportCast Agent Security Extension and ReportCast Server Security Extension, you can override the default authentication process and customize the user name and password that ReportCast uses to access the e.Reporting Server. If you enable RCASE, ReportCast Agent calls it to authenticate users each time it receives a request to connect to the e.Reporting Server. If you enable RCSE, ReportCast Server calls it to authenticate users each time it receives a request to connect to the e.Reporting Server.

RCASE can access HTTP headers and the HTTP body. You can customize RCASE to retrieve the HTTP header and body information, perform authentication against a security database, update the HTTP header and body information, and pass the authenticated request to the ReportCast Server.

RCSE can access the HTTP headers, verify authentication information against a security database, modify the HTTP headers, and pass the request to the e.Reporting Server.

Depending upon your site's needs, you can customize authentication for the ReportCast Agent, the ReportCast Server, or both. For example, your ReportCast configuration might have one ReportCast Server communicating with several ReportCast Agents, each with its own security requirements. You can:

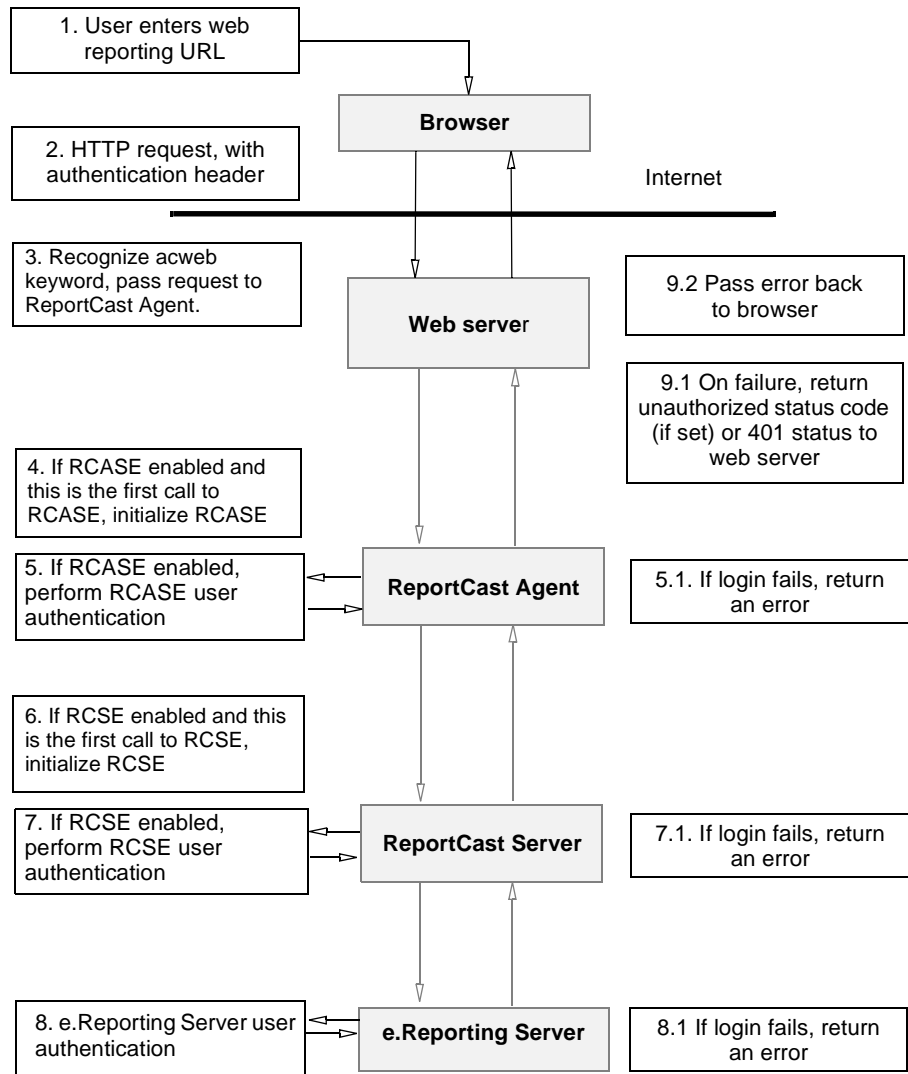
- Create custom security DLL's for each ReportCast Agent.

- Create a custom security DLL for the ReportCast Server that handles custom security requirements depending upon the ReportCast Agent that sent a request.
- Handle authentication only on the ReportCast Agent.
- Handle authentication only on the ReportCast Server.

You enable the ReportCast Security Extension by creating registry entries in NT or by setting environment variables on UNIX. You must create the registry entries or set the environment variables for each ReportCast Agent Security Extension or ReportCast Server Security Extension. The registry entries or environment variables must reside on the same machine with the ReportCast Agent Security Extension or ReportCast Server Security Extension to which they apply.

The following illustration shows the authentication process using the ReportCast Security Extension.

The following illustration shows an overview of the authentication process using the ReportCast Agent and ReportCast Server Security Extensions.

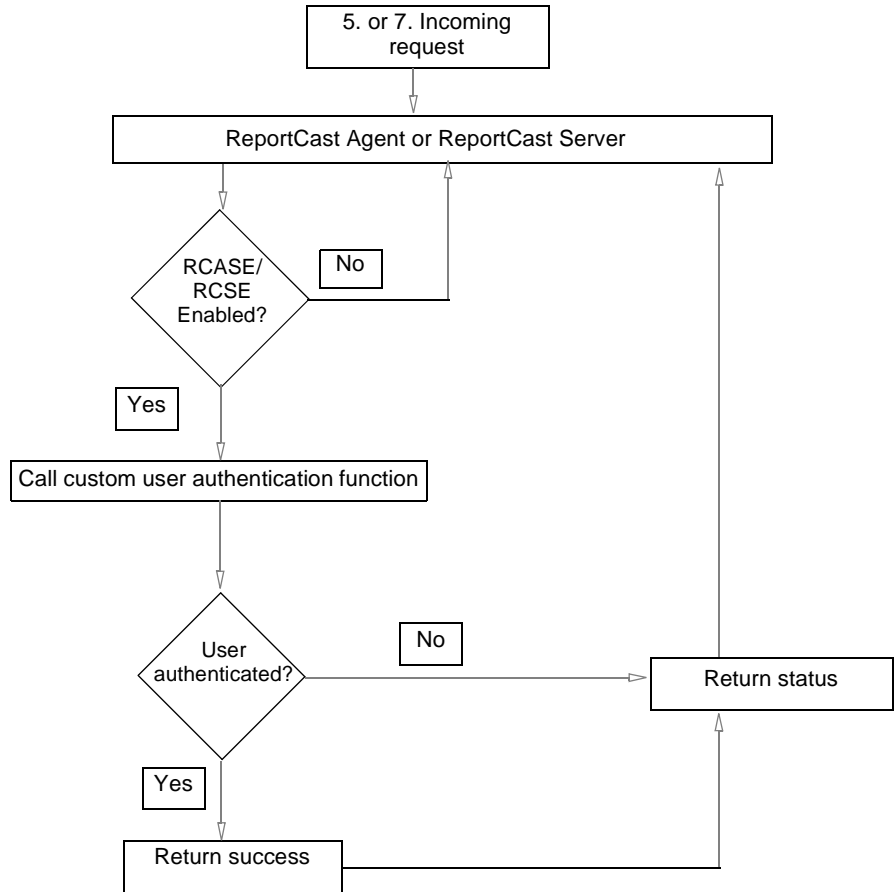


The process is similar to the default authentication process:

- 1 The user enters the web reporting home page URL in the browser.
- 2 The browser passes the HTTP request to the web server, including the authentication header.
- 3 The web server recognizes the acweb keyword and passes the request information to ReportCast Agent.

- 4 If the ReportCast Agent Security Extension is enabled, and this is the first call to RCASE, ReportCast initializes RCASE.
- 5 Authenticate the user using your custom security library (on UNIX) or DLL (on NT).
 - 1 If authentication fails, return an error to ReportCast Agent.
 - 2 If authentication succeeds, pass the request to ReportCast Server.
- 6 If the ReportCast Server Security Extension is enabled, and this is the first call to RCSE, ReportCast initializes RCSE.
- 7 Authenticate the user using your custom security library (on UNIX) or DLL (on NT).
 - 1 If authentication fails, return an error to ReportCast Server.
 - 2 If authentication succeeds, pass the request to the e.Reporting Server.
- 8 The e.Reporting Server processes the user login and password.
 - 1 If login fails, the e.Reporting Server returns an error to ReportCast.
 - 2 If login succeeds, the e.Reporting Server returns successful to ReportCast.
- 9 If login failed:
 - 1 ReportCast returns the `AC_REPORTCAST_UNAUTHORIZED_STATUS_CODE`, if set, or the HTTP 401 status code to the web server.
 - 2 The web server returns the error status to the browser.
 - 3 The browser displays the Login dialog box, and the process begins again.

The following diagram shows the ReportCast Agent and ReportCast Server Security Extension authentication process in more detail.



Developing the ReportCast Security Extension

To override the default authentication process and enable the ReportCast Agent Security Extension or ReportCast Server Security Extension, you develop a custom DLL to contain the RCASE or RCSE security library and set a registry key on NT or an environment variable on UNIX.

ReportCast Agent Security Extension library functions

When the ReportCast Agent Security Extension is enabled, ReportCast calls the RCASE library functions:

RCASE function	Description
AcRCAgentAuthenticateRequestHeader	Authenticate or filter the HTTP request headers and put the filter data into a buffer.
AcRCAgentAuthenticateRequestBody	Authenticate the request data.

If AcRCAgentAuthenticateRequestHeader does not modify the HTTP request headers or if RCASE is disabled, ReportCast Agent sends all the HTTP request headers to ReportCast Server.

If AcRCAgentAuthenticateRequestBody does not modify the HTTP request data or if RCASE is disabled, ReportCast Agent sends the original request body to ReportCast Server.

AcRCAgentAuthenticateRequestBody

Authenticates the HTTP request data. Call this function after you call AcRCAgentAuthenticateRequestHeader.

Syntax

```
int AcRCAgentAuthenticateRequestBody ( const char *httpRequest, char  
    *filteredRequest, int *size )
```

const char *httpRequest

Input. The request data sent by the web browser.

char *filteredRequest

Output. The filtered request modified by AcRCAgentAuthenticateRequestBody().

int *size

Input/Output. The size of the buffer filteredHttpRequest. AcRCAgentAuthenticateRequestBody modifies the value of size to indicate the actual size of the filtered HTTP headers.

The following are the values for size.

Value of size	Description
0	The request body was not modified.
<= size	The new request body is in FilteredHttpRequest and the value of the parameter size is the length of FilteredHttpRequest.
> size	The function requires a larger buffer.

Description

Authenticate or filter the request data and place the filtered data into the buffer filteredHttpRequest. If AcRCAgentAuthenticateRequestBody does not modify the request body, ReportCast sends the original request body to ReportCast Server.

Return values

AcRCAgentAuthenticateRequestBody returns the following values:

Return value	Description
0	Success. ReportCast Agent can send the data to ReportCast Server.
1	Failure. ReportCast Agent sends an error page to the web browser. ReportCast uses either the 401 status code, or the value of the NT registry key or UNIX environment variable AC_REPORTCAST_UNAUTHORIZED_STATUS_CODE.
2	Increase the buffer size. The parameter size contains the new value for the size of the filteredHttpHeaders buffer. ReportCast Agent allocates filteredHttpHeaders with the value of size and repeats the request. The new size must be larger than the original size.

AcRCAgentAuthenticateRequestHeader

Authenticates or filters the HTTP request headers and places the filtered headers into the buffer filteredHttpHeaders. Call this function before you call AcRCAgentAuthenticateRequestBody.

Syntax

```
int AcRCAgentAuthenticateRequestHeader ( const char **httpHeaders, int
    *numberOfHeaders, char *filteredHttpHeaders, int *size )
```

const char **httpHeaders

Input. An array of null-terminated strings, each representing an HTTP header.

int *numberOfHeaders

Input/Output. Used for input, numberOfHeaders contains the number of elements in the httpHeaders array. Used for output, numberOfHeaders contains the number of headers stored in the buffer FilteredHttpHeaders.

char *filteredHttpHeaders

Output. A buffer containing the filtered headers separated by a null character.

int *size

Input/Output. The size of the buffer filteredHttpHeaders. Used for input, size contains the size of filteredHttpHeaders. Used for output, size contains one of the following values:

Value of size	Description
0	The request headers were not modified.
<= size	The new request headers are in FilteredHttpHeaders and the value of the parameter size is the length of FilteredHttpHeaders.
> size	The function requires a larger buffer.

Description

Authenticate or filter the request headers and place the filtered headers into the buffer filteredHttpHeaders. If AcRCAgentAuthenticateRequestHeader does not modify the request headers, ReportCast sends the original request headers to ReportCast Server.

Return values

AcRCAgentAuthenticateRequestHeader returns the following values:

Return value	Description
0	Success. ReportCast Agent can send the data to ReportCast Server.
1	Failure. ReportCast Agent sends an error page to the web browser. ReportCast uses either the 401 status code, or the value of the NT registry key or UNIX environment variable AC_REPORTCAST_UNAUTHORIZED_STATUS_CODE.
2	Increase the buffer size. The parameter size contains the new value for the size of the filteredHttpHeaders buffer. ReportCast Agent allocates filteredHttpHeaders with the value of size and repeats the request. The new size must be larger than the original size.

Example

```
#ifdef WIN32
#include <windows.h>
#define EXPORT WINAPI
#else
#define EXPORT
#endif

#ifdef __cplusplus
extern "C" {
#endif

int EXPORT
AcRCAgentAuthenticateRequestHeader(const char ** httpHeaders, int*
numberOfHeaders, char* filteredHttpHeaders, int* size) {

/* Your code here */

}

int EXPORT
AcRCAgentAuthenticateRequestBody(const char *httpRequest, char*
filteredHttpRequest, int* size) {

/* Your code here */

}
```

```
#ifdef __cplusplus
}
#endif
```

ReportCast Server Security Extension library functions

When the ReportCast Server Security Extension is enabled, ReportCast calls the RCSE library functions:

RCSE function	Description
AcWebSecurityStart	Initialize the RCSE library.
AcWebSecurityAuthenticate	Authenticate a user by translating user security information into e.Reporting Server login information.
AcWebSecurityStop	Shut down the RCSE library.

If the AcWebSecurityStart() or AcWebSecurityStop() functions are not present in the RCSE library, ReportCast continues and does not issue an error message. AcWebSecurityAuthenticate() must be present in the RCSE library if the ReportCast Security Extension is enabled.

For more information about enabling the ReportCast Security Extension, see “Enabling RCASE and RCSE,” later in this chapter.

For information about setting the security variables, see “Setting the ReportCast security variables,” earlier in this chapter.

AcWebSecurityStart function

Initializes the ReportCast security library.

Syntax

```
void AcWebSecurityStart( )
```

Description

Performs internal initialization, such as loading tables. If AcWebSecurityStart() is present in the RCSE library, the ReportCast calls it once, when it loads the security library.

AcWebSecurityAuthenticate function

Authenticates a user.

Syntax

```
int AcWebSecurityAuthenticate(char **httpHeaders, int numberOfHeaders,  
    char *username, char *password)
```

Description

Called each time ReportCast connects to an e.Reporting Server to service a request. If this function is not present in the RCSE library, or an error occurs during the operating system call to locate the function, ReportCast sends an HTML error page generated from the error.achtml template to the browser.

This function supports using the HTTP headers from the browser to determine the user name and password that ReportCast uses to log on to the e.Reporting Server.

httpHeaders [in]

An array of null-terminated strings, each representing an HTTP header using the format defined in the HTTP 1.1 specification.

numberOfHeaders [in]

Number of elements in the httpHeaders array.

username [out]

Name ReportCast uses to log on to the e.Reporting Server. ReportCast is responsible for allocating and freeing this memory. This memory can hold a string 128 bytes long. Your code must copy the user name into this location.

password [out]

Password to use to log on to the e.Reporting Server. ReportCast is responsible for allocating and freeing this memory. This memory can hold a string 128 bytes long. Your code must copy the password into this location.

Return value

0 on success, 1 on failure.

If AcWebSecurityAuthenticate() returns 1, ReportCast does not attempt to log on to the e.Reporting Server and returns a status code of 401 or the value

specified by the `AC_REPORTCAST_UNAUTHORIZED_STATUS_CODE` variable to the web browser.

AcWebSecurityStop function

Shuts down the ReportCast security library.

Syntax

```
void AcWebSecurityStop( )
```

Description

Performs internal cleanup for the RCSE library. If `AcWebSecurityStop()` is present in the library, it is called once, when ReportCast shuts down.

Development considerations

Here are some items to consider for developing the Actuate ReportCast Security Extension:

- The functions that the security library exports, `AcWebSecurityStart()`, `AcWebSecurityAuthenticate()`, and `AcWebSecurityStop()`, should all be declared extern C if you are writing the library in C++. This procedure prevents the compiler from changing the symbol name into something other than what ReportCast is looking for.
- Your security library must be thread-safe and cannot depend on any one thread handling a particular request.
- The HTTP header names that are passed to the security library should be in the format specified in the HTTP specification. The headers do not use the CGI or ISAPI format. For example, the header `Accept-Charset` is used, rather than `HTTP_ACCEPT_CHARSET`. This practice allows you to write the Security Extension code using the standard HTTP headers so that it is portable among different web servers and platforms.
- Ensure that your code is not case-sensitive.
- When testing your ReportCast Security Extension library, be aware that ReportCast caches connections to the e.Reporting Server. If you are changing user passwords on the server during your testing, you may have to clear cached connections using the ReportCast administration page.

Example

The following code example shows a template used to develop a security library:

```
#ifndef WIN32
#include <windows.h>
#endif

#ifdef __cplusplus
extern "C" {
#endif

void
#ifdef WIN32
WINAPI
#endif
AcWebSecurityStart() {
// Your code here.
}

void
#ifdef WIN32
WINAPI
#endif
AcWebSecurityStop() {
// Your code here.
}

int
#ifdef WIN32
WINAPI
#endif
AcWebSecurityAuthenticate(char **httpHeaders,
    int numberOfHeaders,
    char *username,
    char *password) {
// Your code here.
}

#ifdef __cplusplus
}
#endif
```

NT-specific information

On NT, you must use the WINAPI calling convention in the function definitions for the RCSE functions. Declare the RCSE functions in a DEF file that is linked to the DLL.

For example, the following code shows the contents of a DEF file to link into the DLL:

```
DESCRIPTION 'Your Security Extension Description'
EXPORTS
    AcRCAgentAuthenticateRequestBody
    AcRCAgentAuthenticateRequestHeader
    AcWebSecurityStart
    AcWebSecurityStop
    AcWebSecurityAuthenticate
```

Server-specific information

The following sections contain additional information that applies to the supported web servers, Netscape and Microsoft.

Netscape server

The ReportCast Agent Security Extension supersedes any previous ReportCast authentication customization solutions. The security extension feature allows implementation of custom security authentication that is portable and avoids the need to add directives to the obj.conf file to achieve customization.

Microsoft Internet Information server

By default, the Microsoft Internet Information server (IIS) uses the Windows NT security information from the browser to grant or deny access to all resources on the web server. IIS considers the ReportCast DLL to be the resource to which it must authorize access, rather than the e.Reporting Server. Each user name and password must exist on both the NT system to access ReportCast, and the e.Reporting Server to access the Report Encyclopedia. Since this setup is not practical in some cases, the ReportCast configuration customizes this behavior.

If the security extension is not enabled, ReportCast overrides the default behavior of IIS and a component of ReportCast acts as an authentication filter. This filter instructs IIS to access ReportCast as the NT user designated for IIS anonymous access. In this configuration, IIS must be configured to allow anonymous access and Windows NT Challenge/Response must be disabled. When ReportCast attempts to connect to the e.Reporting Server, it uses Basic Authentication to gather the user name and password information.

The alternative authentication method is to use the ReportCast Agent Security Extension. If the security extension is enabled, ReportCast does not override the default behavior of IIS to authorize access to the ReportCast DLL. Authentication information for the ReportCast DLL comes from the browser. ReportCast does not filter the user information. The user information coming from the browser must identify a valid NT user. If the browser does not send authentication information, and if IIS is configured to allow anonymous access, IIS accesses ReportCast as the NT user designated for anonymous access.

If you want to access ReportCast using a different Windows NT user than that specified by the browser, install an additional ISAPI filter DLL in Internet Information Server.

Depending on the security model you use, the security extension returns the same user information as the browser sent, or maps the user information to a completely different user. In either case, the user it returns must exist in the e.Reporting Server if ReportCast is to log on successfully.

Enabling RCASE and RCSE

Set `AC_REPORTCAST_SECURITY_LIBRARY` for each RCASE and RCSE for which you created a security library. Setting `AC_REPORTCAST_SECURITY_LIBRARY` allows ReportCast Agent and ReportCast Server to locate your security libraries. If this variable is not set, ReportCast Agent and ReportCast Server use the default authentication scheme.

The variable value is the name of the DLL or shared object file that contains the authentication functions. The name can include either an absolute or relative path.

ReportCast loads the library once, the first time a request is made to ReportCast. ReportCast unloads the library when it exits. If the library cannot be loaded, ReportCast sends an error page to the browser.

You must enable each RCASE and RCSE. If you have multiple ReportCast Agent Security Extensions, enabling RCASE on one machine enables only that RCASE and not any others. If you have multiple ReportCast Server Security Extensions, enabling RCSE on one machine enables only that RCSE and not any others.

Use `regedit` to enable RCASE and RCSE on NT. For more information about running `regedit` and creating registry keys, see “Setting the ReportCast security variables,” earlier in this chapter.

Setting the RCASE security variable

On each NT machine on which an RCASE resides:

- 1 Run regedit and navigate to HKEY_LOCAL_MACHINE\Software\Actuate\Actuate ReportCast Agent\5.
- 2 Create a new registry key, AC_REPORTCAST_SECURITY_LIBRARY.
- 3 Set the AC_REPORTCAST_SECURITY_LIBRARY variable to the full pathname of the ReportCast Agent Security Extension library's location.

On each UNIX machine on which an RCASE resides, type:

```
setenv AC_REPORTCAST_SECURITY_LIBRARY <RCASE library location>
```

where RCASE library location is the full pathname of the ReportCast Agent Security Extension library's location.

RCASE is now enabled.

Setting the RCSE security variable

On each NT machine on which an RCSE resides:

- 1 Run regedit and navigate to HKEY_LOCAL_MACHINE\Software\Actuate\Actuate ReportCast Server\5.
- 2 Create a new registry key, AC_REPORTCAST_SECURITY_LIBRARY.
- 3 Set the AC_REPORTCAST_SECURITY_LIBRARY variable to the full pathname of the ReportCast Security Extension library's location.

On each UNIX machine on which an RCSE resides, type:

```
setenv AC_REPORTCAST_SECURITY_LIBRARY <RCSE library location>
```

where RCSE library location is the full pathname of the ReportCast Security Extension library's location.

RCSE is now enabled.

Security for Internet deployment

This chapter contains the following topics:

- About network security and Actuate
- Data protection strategies
- Understanding Actuate component communication
- Deploying Actuate products in Internet environments

About network security and Actuate

This chapter discusses configuring Actuate e.Reporting Suite 5 products for deployment in a secured network environment. Project managers, installation engineers, and security team members can plan hardware and software requirements, installation configuration options, and how Actuate fits into the Internet security infrastructure of their organization based on information in this chapter.

The topics discussed include the technologies that deter unauthorized access to corporate data, such as firewalls, Network Address Translation (NAT), and proxy servers.

This chapter is not a guide to securing your network. This chapter is a general guide to how Actuate products work with secure environments. This chapter assumes a working knowledge of TCP/IP communications.

Data protection strategies

Actuate e.Reporting Suite 5 provides a structured content generation solution for web applications. Deploying Actuate applications developed for the Internet requires planning for network security.

Internet applications support access to information within an organization from outside that organization. Because the organization's internal network is connected to the Internet, there is the risk of unauthorized access to the corporate network and to the data that resides on that network.

Protecting corporate data

Typically companies use firewalls to prevent unauthorized access to corporate networks and data.

A firewall is a system or group of systems that restrict access between two networks, such as an organization's internal network and the Internet. Firewalls keep unauthorized users out. As a result, firewalls prevent damage caused by malicious programs such as worms and viruses from spreading to other parts of your network. At the same time, firewalls allow legitimate business to tunnel through the firewall and be efficiently conducted on your network.

Firewalls can be used to restrict access between two internal networks, for example, the accounting and engineering networks. Security teams configure firewalls to allow traffic using specific protocols, such as HTTP, over specific network addresses and ports.

Companies also use Network Address Translation (NAT). NAT is routers and software that allow private networks using unregistered, private IP (Internet Protocol) addresses to connect to the Internet.

Finally, proxy servers, specialized web servers or hardware that operate on or behind a firewall, improve efficient use of network bandwidth and offer better network security.

Organizations use one or a combination of these technologies to prevent unauthorized access to the corporate network.

Understanding Actuate component communication

This section describes the relationship between Actuate component communication and your secure network environment.

Actuate e.Reporting Server, ReportCast, and all client products use TCP/IP and related protocols for all network communication. The communications process between client products and the e.Reporting Server is the same as the communications process between ReportCast and the e.Reporting Server.

The originating or outbound port is the port that a component uses to initiate two-way connections between itself and another component.

The destination or inbound port is the port to which an originating port initiates two-way connections.

Actuate components have these communications processes:

- Communicating between the web browser and web server
- Establishing the e.Reporting Server port
- Connecting to and browsing the Report Encyclopedia
- Running e.reports
- Viewing e.reports in DHTML format
- Viewing e.reports using the LRX

You configure the ports to use for specific communications tasks at e.Reporting Server and ReportCast startup using registry settings on NT systems or environment variables on UNIX systems.

The types of ports you can configure are:

- e.Reporting Server port. This port is used for the ReportCast to e.Reporting Server connection.
- Cached connection management. This port is used to control the number of ReportCast's cached connections.

- View process port. This port is used for viewing DHTML reports, PDF content, and XML data and XML display.
- Data reply connection. This port is used to view reports using the LRX or Viewer, retrieving parameters for running reports, and notifications for report requests.

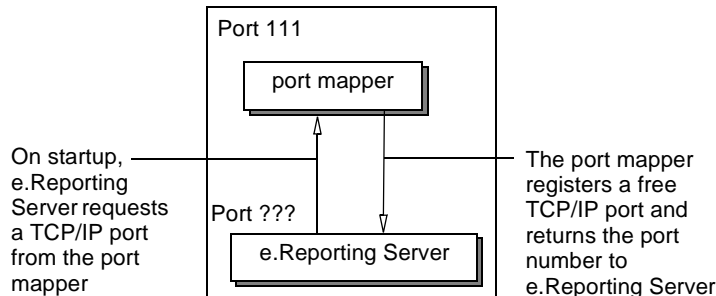
Establishing the e.Reporting Server port

The e.Reporting Server port is the destination port for connections from ReportCast to the e.Reporting Server. ReportCast and the e.Reporting Server use the connection for:

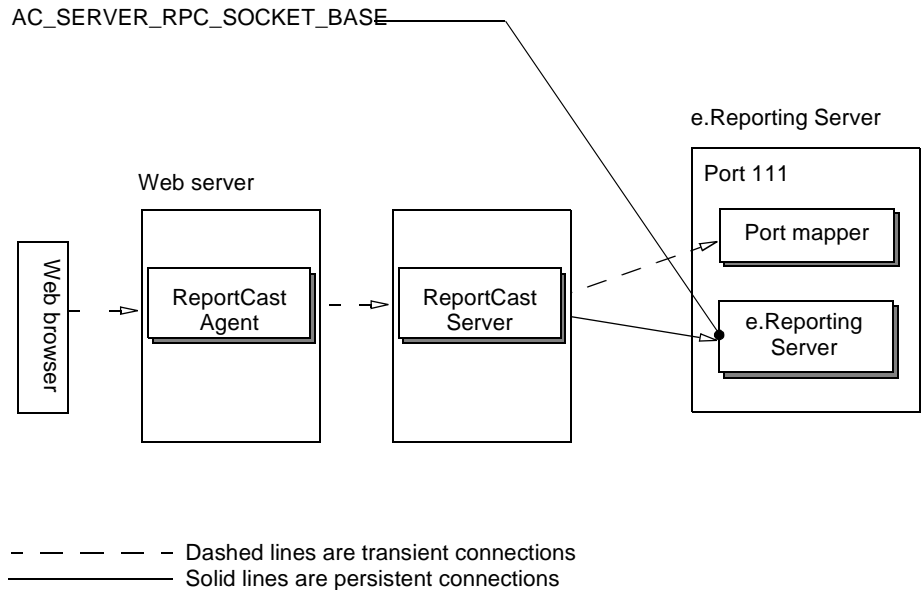
- Login
- Report Encyclopedia browsing
- Report Encyclopedia administration
- Initiating report requests
- Initiating report viewing

e.Reporting Server applications use the TCP/IP port mapper to determine the port through which to communicate with the e.Reporting Server. When the e.Reporting Server starts, it requests a communications port from the port mapper. The port mapper locates an available TCP/IP port, registers the port number, and returns the port number to the e.Reporting Server.

The port mapper listens for queries on port 111. Applications query the port mapper for the port used by the e.Reporting Server. The port mapper responds with the registered port. Applications use the registered port to connect to the e.Reporting Server.



The following illustration shows the connection between ReportCast and the e.Reporting Server.



The e.Reporting Server port uses the RPC protocol over TCP. There is one destination port for all user connections. You can configure the number of cached connections. The peak number of connections depends on the user load.

Configuring the e.Reporting Server port

You can set the port number to use for connections and the number of ports to search if that port is not available.

To configure the e.Reporting Server port, set the following registry keys on NT or environment variables on UNIX on the e.Reporting Server machine:

- AC_SERVER_RPC_SOCKET_BASE sets the port number that the e.Reporting Server attempts to register with the port mapper.
- AC_SERVER_RPC_SOCKET_COUNT sets the number of ports search if the registered port is not available.

When the e.Reporting Server attempts to register a port with the port mapper, it searches from AC_SERVER_RPC_SOCKET_BASE to AC_SERVER_RPC_SOCKET_BASE + AC_SERVER_RPC_SOCKET_COUNT-1 to find an open port. If no ports are available within the configured range, the port mapper allocates a port. The default is to allow the port mapper to allocate the port.

For detailed information about configuring registry keys on NT or environment variables on UNIX, see Chapter 2, “Report server administration on UNIX,” or Chapter 3, “Report server administration on Windows servers,” in the *Actuate e.Reporting Server Guide*.

Configuring cached connection management

ReportCast caches connections to the e.Reporting Server port, AC_SERVER_RPC_SOCKET_BASE. To control the number of cached connections, set the following registry keys on NT or environment variables on UNIX on the web server machine:

- AC_REPORTCAST_CONNECTION_CACHE_SIZE limits the total number of connections. Actuate drops the oldest connections automatically.
- AC_REPORTCAST_CONNECTION_CACHE_TIMEOUT limits the life of the connection.

For more information about configuring cached connections, see Chapter 2, “Administering ReportCast.”

Web browser to web server communication

Web browser to web server communication uses the standard HTTP or HTTPS (SSL) protocols. Actuate products do not require any special configuration or consideration for HTTP or HTTPS communication. HTTPS is transparent to Actuate.

Connecting to and browsing the Report Encyclopedia

The following refers to ReportCast, rather than ReportCast Agent and ReportCast Server, in order to simplify discussion. The ReportCast process is:

- 1 A user makes a request using a web browser.
- 2 The web browser passes the request to the web server.
- 3 The web server recognizes the acweb keyword in the request and passes it to ReportCast Agent.
- 4 ReportCast Agent passes the request to ReportCast Server.
- 5 ReportCast Server processes the request for the e.Reporting Server and passes it to the e.Reporting Server.
- 6 The e.Reporting Server processes the request and sends the results to ReportCast Server.

- 7 ReportCast Server prepares a web page with the results and sends it to ReportCast Agent.
- 8 ReportCast Agent sends the web page to the web server.
- 9 The web server passes the web page to the web browser.

ReportCast Agent and ReportCast Server may also call their security extensions to authenticate user access to the e.Reporting Server, depending upon the site's configuration and requirements. See Chapter 6, "Internal ReportCast security," for more information about the ReportCast security extensions.

The process for connecting to and browsing the Report Encyclopedia is:

- The web browser connects to the web server using HTTP or HTTPS to a standard port, usually port 80, with a request to view the list of files in a folder.
- On the first connection, ReportCast opens an originating port.
- ReportCast connects to the port mapper using TCP to determine the e.Reporting Server port.
- The port mapper returns the e.Reporting Server port.
- ReportCast connects to the e.Reporting Server using the destination port supplied by the port mapper. ReportCast uses the new connection to retrieve information from the e.Reporting Server. All users share one destination port.
- ReportCast generates the web reporting page and returns it to the web server.
- The web server returns the web reporting page to the web browser using HTTP.

Connections from the web browser to the web server and ReportCast, and from ReportCast to the port mapper, are transient. The connection between ReportCast and the e.Reporting Server remains open.

The desktop products use the same process to communicate with the e.Reporting Server.

Running and viewing reports

This section describes the process of running and viewing reports and how to configure the view process and data reply ports.

Running reports

One connection is used for each concurrent user running a report. Each connection opens a pair of ports: one on the originating side, one on the destination side. Data moves in both directions.

The process for running reports is:

- The web browser connects to the web server using HTTP or HTTPS to a standard port, usually port 80.
- ReportCast requests the parameters needed to run the report.
- The e.Reporting Server establishes a new connection to ReportCast for sending the report's parameter data.
- ReportCast generates the web reporting page and returns it to the web server.
- The web server returns the web reporting page to the web browser using HTTP.

The desktop products use the same process to run a report.

The process for running reports uses data reply ports. Configuring data reply ports is described in "Configuring data reply ports," later in this chapter.

Configuring data reply ports

The LRX and Viewer use the data reply ports to view reports, to retrieve parameters for running reports, and for report request notification.

The data reply connection process is:

- ReportCast logs in and gets an RPC connection to the e.Reporting Server.
- User initiates a request to run a report or to view content using the LRX viewer.
- ReportCast requests parameters from the e.Reporting Server or requests content for the LRX viewer.
- e.Reporting Server acknowledges the request and initiates a connection to ReportCast:
 - e.Reporting Server opens an originating port (AC_SERVER_REPLY_SOCKET_BASE).
 - e.Reporting Server connects to the destination port on the ReportCast machine (AC_SERVER_SOCKET_BASE).
- e.Reporting Server sends the report data.
- e.Reporting Server closes the connection.

To configure originating data reply ports, set the following registry keys on NT or environment variables on UNIX on the e.Reporting Server machine:

- `AC_SERVER_REPLY_SOCKET_BASE` sets the number at which to begin allocating originating ports.
- `AC_SERVER_REPLY_SOCKET_COUNT` sets the maximum number of ports to use.

When the e.Reporting Server receives a data reply request, it searches from `AC_SERVER_REPLY_SOCKET_BASE` to `AC_SERVER_REPLY_SOCKET_BASE + AC_SERVER_REPLY_SOCKET_COUNT-1` to find an open port. The default value for `AC_SERVER_REPLY_SOCKET_BASE` is 4096. The default value for `AC_SERVER_REPLY_SOCKET_COUNT` is 8192.

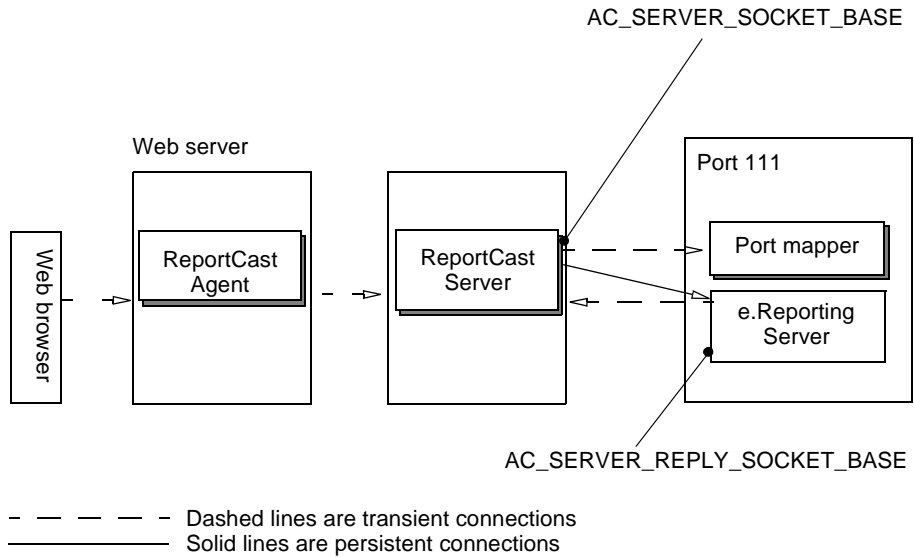
To configure destination ports, set the following registry keys (on NT) or environment variables (on UNIX) on the ReportCast or Viewer machine:

- `AC_SERVER_SOCKET_BASE` sets the number at which to begin allocating destination ports.
- `AC_SERVER_SOCKET_COUNT` sets the maximum number of ports to use.

When the e.Reporting Server receives a data reply request, it searches from `AC_SERVER_SOCKET_BASE` to `AC_SERVER_SOCKET_BASE + AC_SERVER_SOCKET_COUNT-1` to find an open port. The default value for `AC_SERVER_SOCKET_BASE` is 4096. The default value for `AC_SERVER_SOCKET_COUNT` is 8192.

Data reply ports use the TCP protocol and the UDP protocol for notifications. Actuate opens one port for each concurrent data request. For example, if ten users simultaneously request a report, ten pairs of ports open. For each pair of ports, one is on the origination side and one is on the destination side. Data reply connections are opened and closed for each data reply request.

The following illustration shows the connections between ReportCast and the e.Reporting Server for data reply requests.



Viewing e.reports as DHTML, PDF, or XML

Viewing DHTML, PDF, or other structured content using the view process uses one port per view process. Each view process port is shared by all users.

The process for viewing e.reports in DHTML format is:

- The web browser connects to the web server using HTTP or HTTPS to a standard port, usually port 80, with a request to view a report as DHTML.
- ReportCast makes a request to view a DHTML report.
- The e.Reporting Server tells ReportCast which port to use to retrieve the DHTML content.
- ReportCast connects to the e.Reporting Server port for viewing DHTML content. The e.Reporting Server returns DHTML content to ReportCast. Actuate uses one port for each view process. Typically, the number of ports used is one or two. See the *Actuate e.Reporting Server Guide* for more information. All users share the same view process ports.
- ReportCast generates the web reporting page and returns it to the web server.
- The web server returns the web reporting page to the web browser using HTTP.

The process of viewing e.reports in DHTML, PDF, or XML uses the view process port. Configuring the view process port is described in “Configuring the view process port,” later in this chapter.

Configuring the view process port

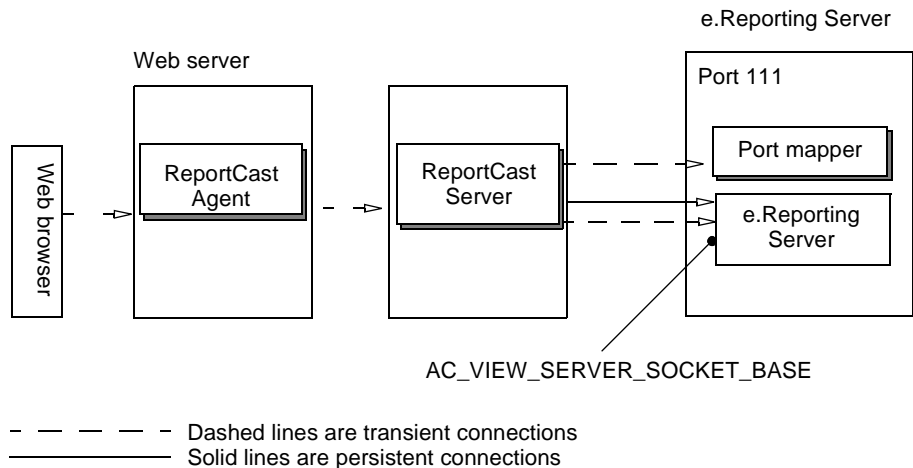
The view process port is used for viewing DHTML, PDF, and XML content. Set the following registry keys on NT or environment variables on UNIX on the e.Reporting Server machine:

- `AC_VIEW_SERVER_SOCKET_BASE` sets the port number at which to begin allocating view process connection ports. The default value is 15500.
- `AC_VIEW_SERVER_SOCKET_COUNT` sets the maximum number of ports to search for open ports. The default value is 200.

When the e.Reporting Server receives a request for the view process, it searches from `AC_VIEW_SERVER_SOCKET_BASE` to `AC_VIEW_SERVER_SOCKET_BASE + AC_VIEW_SERVER_SOCKET_COUNT - 1` to find an open port. Set `AC_VIEW_SERVER_SOCKET_BASE` to a number greater than 1024. Set `AC_VIEW_SERVER_SOCKET_COUNT` to a number less than 65535.

The view process port uses the TCP protocol. Actuate uses one port for each view process. View process connections are opened and closed for each request to view DHTML, XML, or PDF content.

The following illustration shows the connections among ReportCast, the view process, and the e.Reporting Server.



Viewing e.reports using the LRX

One connection is used for each concurrent user viewing a report. Each connection opens a pair of ports: one on the originating side, one on the destination side. Data moves in both directions.

The process for viewing e.reports using the LRX is:

- The web browser connects to the web server using HTTP or HTTPS to a standard port, usually port 80, with the request to view the reporting using the LRX.
- ReportCast makes a request to view a report using the LRX.
- The e.Reporting Server establishes a new connection to a ReportCast port for sending report data for viewing.
- ReportCast generates the web reporting page and returns it to the web server.
- The web server returns the web reporting page to the web browser using HTTP.

The Viewer and other desktop products use the same process to view a report on the e.Reporting Server.

The process for viewing e.reports using the LRX uses data reply ports. Configuring data reply ports is described in “Configuring data reply ports,” earlier in this chapter.

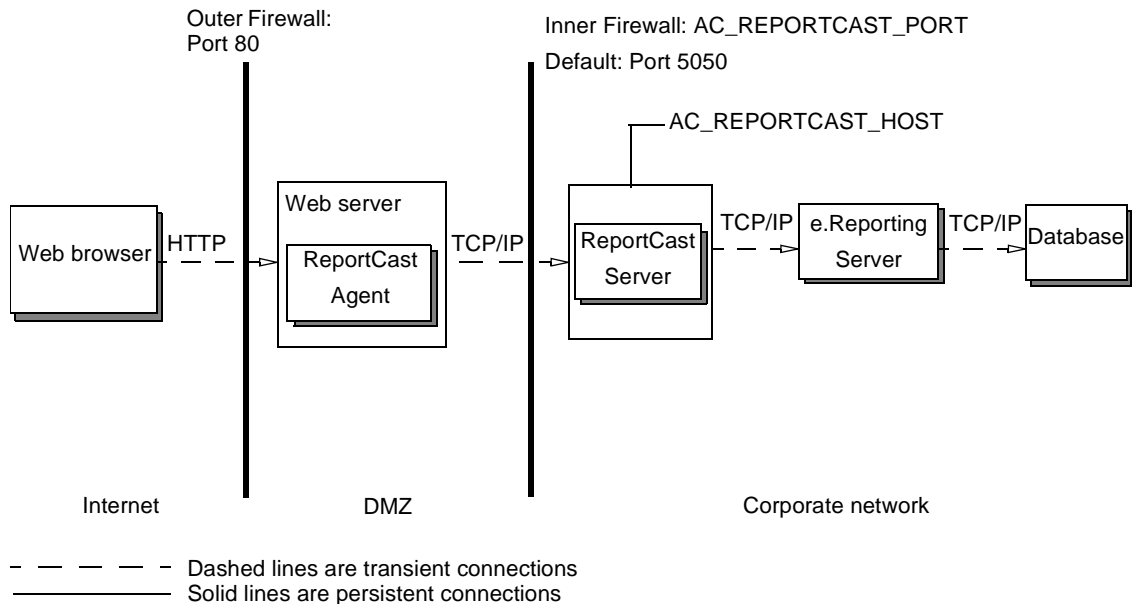
Connecting to the Report Encyclopedia with the split ReportCast configuration

The split ReportCast configuration supports installing the ReportCast Agent on one machine, and the ReportCast Server on another machine. You can simplify the configuration of your firewall by placing the ReportCast Agent with the web server in the DMZ and the ReportCast Server on a separate machine on the corporate net.

The split ReportCast configuration provides the following advantages:

- Requires only one port through the inner firewall. The default is port 5050. The port is configurable.
- Does not require a proxy server.

The following illustration shows the split ReportCast configuration.



The split ReportCast communication process is:

- The web browser connects to the web server using HTTP or HTTPS to a standard port, usually port 80, with a request to view the list of files in a folder, run a report, or view a report.
- The web server passes the request to the ReportCast Agent.
- The ReportCast Agent connects to the ReportCast Server on the port specified by AC_REPORTCAST_PORT on the machine specified by AC_REPORTCAST_HOST and passes the request to the ReportCast Server.
- The ReportCast Server connects to the e.Reporting Server.
- The ReportCast Server generates the web reporting page and returns it to the ReportCast Agent.
- The ReportCast Agent returns the web reporting page to the web server.
- The web server returns the web reporting page to the web browser using HTTP.

Configuring the split ReportCast configuration

To configure the split ReportCast configuration, set the following environment variables:

- `AC_REPORTCAST_PORT` sets the port to use for communicating between the ReportCast Agent and the ReportCast Server. Set `AC_REPORTCAST_PORT` on both the ReportCast Agent and ReportCast Server machines. The port number must match on both machines.
- `AC_REPORTCAST_HOST` sets the name of the machine hosting the ReportCast Server. Set `AC_REPORTCAST_HOST` only on the ReportCast Agent machine.

When the ReportCast Agent receives a request, it connects to the ReportCast Server on the port specified by the `AC_REPORTCAST_PORT` setting, on the host machine specified by the `AC_REPORTCAST_HOST` setting.

The default value for `AC_REPORTCAST_PORT` is 5050. Set `AC_REPORTCAST_PORT` to the same value on both the ReportCast Agent and the ReportCast Server machines.

Set `AC_REPORTCAST_PORT` as follows:

- For Netscape web servers only, on the ReportCast Agent side, set `AC_REPORTCAST_PORT` in the Netscape start script.
- For all web servers including Netscape on the ReportCast Agent side, set `AC_REPORTCAST_PORT` in the `nph-actuate.cgi` and `ph-actuate.cgi` scripts.
- On the ReportCast Server side, set `AC_REPORTCAST_PORT` in `startreportcast.sh`.

Set `AC_REPORTCAST_HOST` on the ReportCast Agent side only as follows:

- For Netscape web servers only, set `AC_REPORTCAST_HOST` in the Netscape start script.
- For all web servers, including Netscape, set `AC_REPORTCAST_HOST` in the `nph-actuate.cgi` and `ph-actuate.cgi` scripts.

Deploying Actuate products in Internet environments

There are many options for configuring network security. This section presents some scenarios and discusses requirements for deploying Actuate products in secure Internet environments.

Some routers and firewalls perform Network Address Translation (NAT). You can set up NAT to perform one-to-one IP address mapping in both directions. In a connection from point A to point B, NAT must map both A to B, and B to A.

The following are sample network security configuration options:

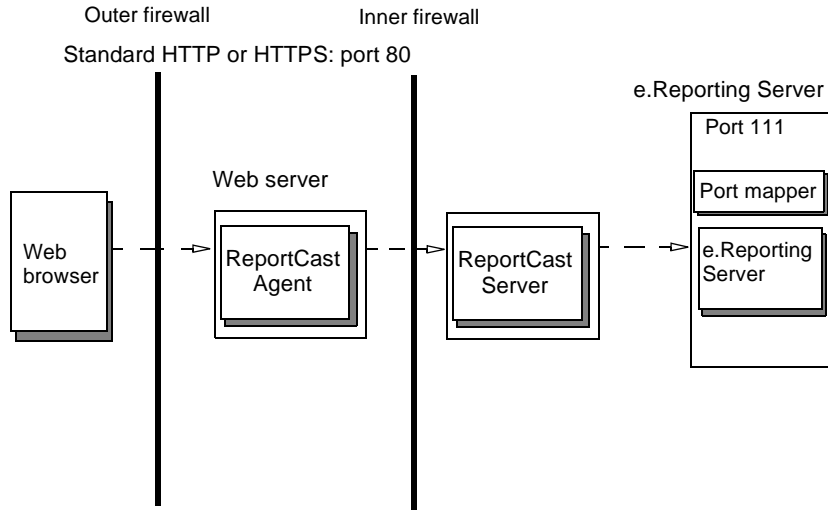
- Split ReportCast installation places the web server and ReportCast Agent in the DMZ. The ReportCast Server and e.Reporting Server reside behind the second firewall. This is the recommended configuration.
- Web server and ReportCast in the DMZ places the web server and ReportCast between two firewalls. The e.Reporting Server resides behind the inner firewall.
- Proxy server in the DMZ places a proxy server between two firewalls. The web server, ReportCast, and e.Reporting Server reside behind the inner firewall.

The following sections discuss firewall configuration options, including port configuration instructions.

Understanding configuration options and firewall setup

Most firewall configurations are set up to accept traffic from specific protocols, such as HTTP or HTTPS traffic only. Generally, corporations expose only limited ports and one IP address to the World Wide Web. All Internet traffic comes through specific ports, usually including port 80, and to the one outside IP address. From port 80, Internet traffic is routed to appropriate internal IP addresses and ports, often using Network Address Translation (NAT).

The web browser to web server communications process is the same for each configuration. If there is a firewall, communication uses standard HTTP or HTTPS protocols, as shown in the following diagram.

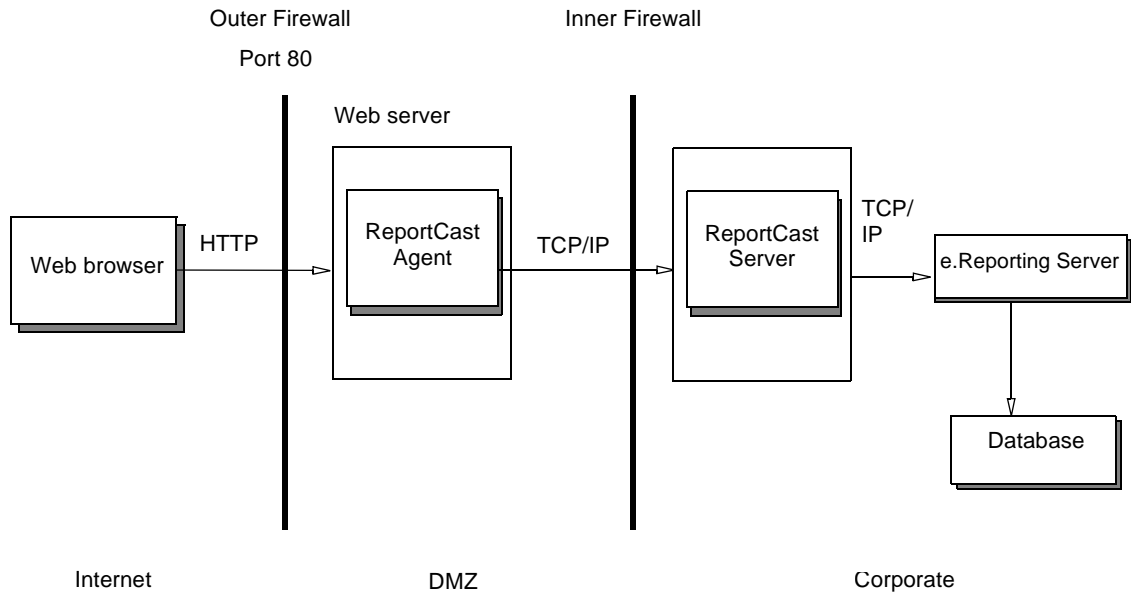


- - - - - Dashed lines are transient connections

The following sections discuss each firewall configuration option.

Dual firewalls with ReportCast Agent in the DMZ

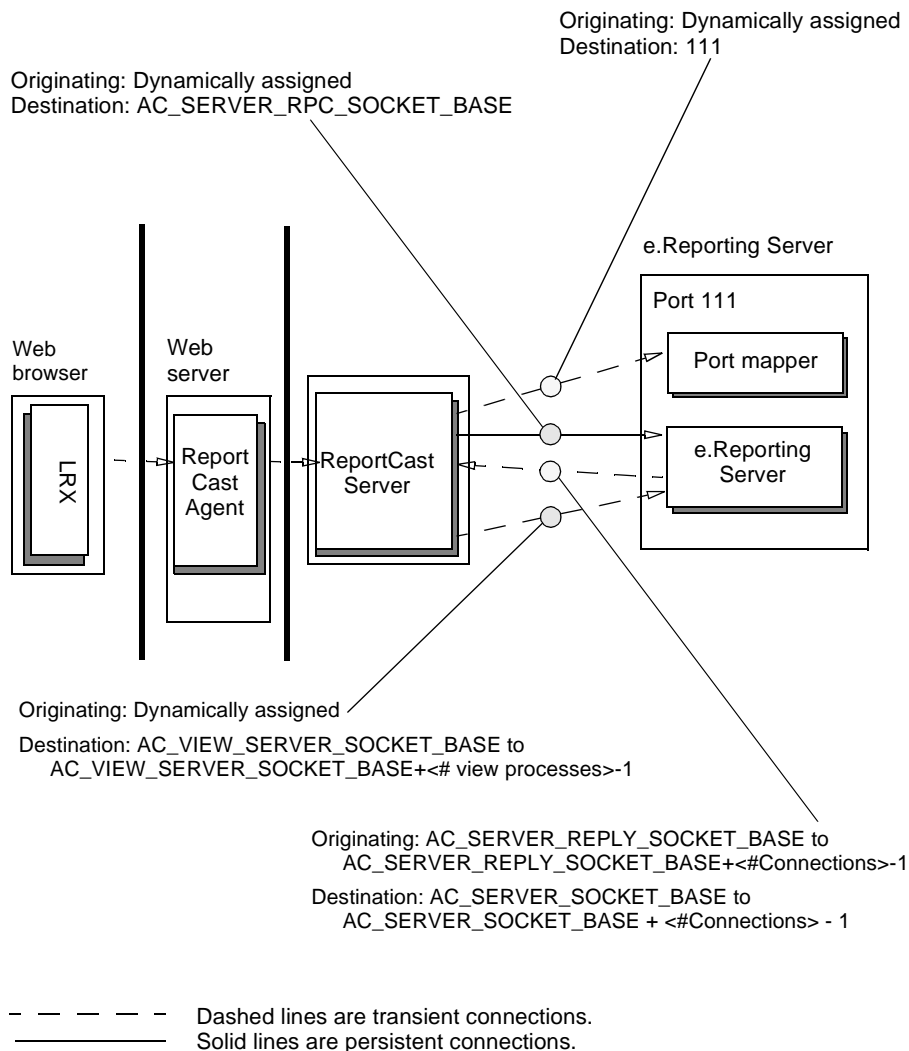
The following illustration shows a dual firewall configuration with the web server and ReportCastAgent in the DMZ.



Web browsers communicate with the web server through one well-known port, usually port 80, in the outer firewall. ReportCast Agent forwards requests through the inner firewall to the corporate net and ReportCast Server. ReportCast Server processes the requests and passes them to the e.Reporting Server using TCP/IP. The e.Reporting Server communicates with the Report Encyclopedia and databases using TCP/IP.

This dual firewall option provides an effective screen against unauthorized access. Internet traffic comes through only one port, generally using only one protocol: HTTP or HTTPS. Users on the Internet cannot gain direct access to the corporate network, because its ports and IP addresses are known only internally. Users on the Internet have limited options for accessing the web server machine, because they can only communicate using one protocol. For example, they cannot access the corporate network using FTP or Telnet.

The following illustration shows the inner firewall configuration.



For information about configuring the originating and destination ports, see “Configuring the e.Reporting Server port,” “Configuring data reply ports,” and “Configuring the view process port,” earlier in this chapter.

NAT considerations for dual firewalls with ReportCast Agent in the DMZ

Actuate supports the NAT Static Address Translation algorithm, where users establish one-to-one connections between corporate (inside) network addresses and global (outside, Internet) network addresses. Actuate

components initiate connections from either side of the firewall or routers. The IP address mapping must operate and be consistent from both sides of the firewall or routers.

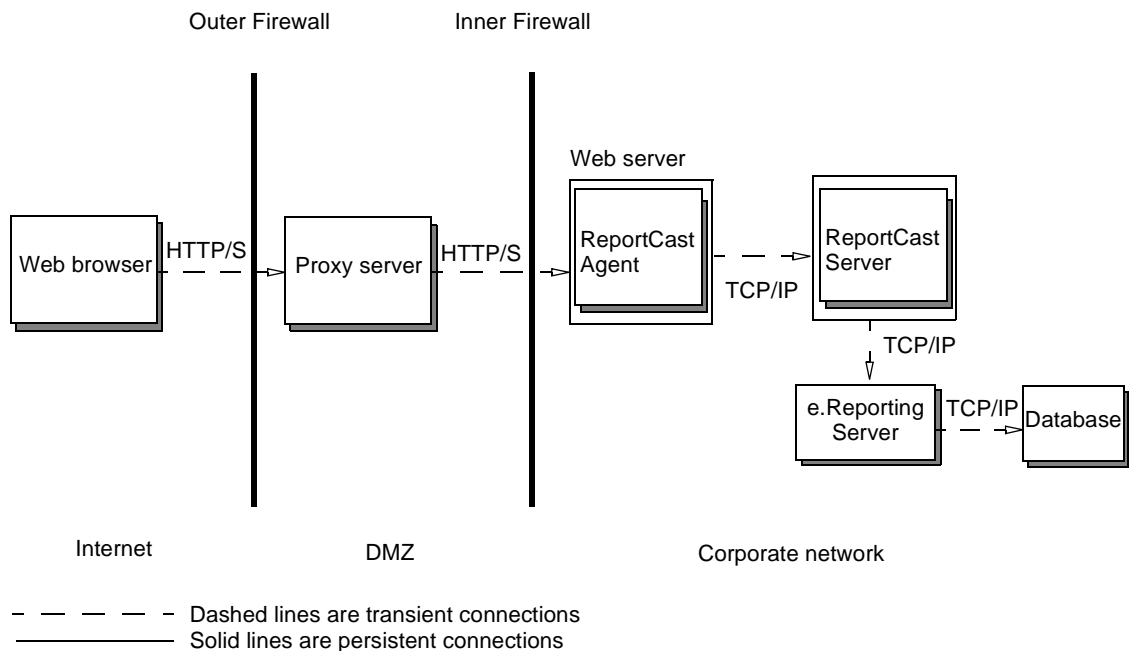
This configuration does not support the following NAT algorithms:

- Dynamic Source Address Translation
- Port Address Translation (PAT)
- Destination Address Rotary Translation

Dual firewalls with a proxy server

In this configuration, a proxy server resides in the DMZ. The web server, ReportCast Agent, ReportCast Server, and e.Reporting Server reside behind the inner firewall on the corporate network. The proxy server accepts HTTP or HTTPS traffic from the Internet, then passes the HTTP or HTTPS requests to the web server on the corporate network. ReportCast receives requests from the web server and passes the requests to the e.Reporting Server using TCP/IP.

The following illustration shows a dual firewall configuration with a proxy server in the DMZ.



The proxy server handles all communications with the Internet using HTTP through one well-known port, usually port 80. The proxy server communicates with the web server on the corporate network using HTTP through one internally-known port. ReportCast and the e.Reporting Server use TCP/IP to communicate.

This configuration option provides one access point for users on the Internet, through one IP address and well-known port, and one access point to the corporate network. The proxy configuration severely limits the possibilities of accessing the corporate network from the Internet.

NAT considerations for dual firewalls with a proxy server

Configure NAT as you usually would for use with a proxy server. Refer to your NAT and proxy server documentation for configuration information.

NAT considerations for the split ReportCast configuration

There are no special NAT considerations for the split ReportCast configuration.

Placing the e. Reporting Server in the DMZ

This configuration is possible, but not recommended. In this configuration, the web server, ReportCast, and the e.Reporting Server reside in the DMZ. The e.Reporting Server contains corporate data in the form of reports. Corporate data is vulnerable to outside attack in this configuration, which presents an unacceptable security risk.

Index

Symbols

as command indicator 119
& command separator 85
* wildcard 123
... wildcard 123
? as directive indicator 80
? wildcard 123

A

AbbreviatedCurrentSize variable 156
AbbreviatedFileSize variable 153
AbbreviatedSize variable 157
AbbreviatedType variable 153
AC_REPORTCAST_VIEWTMPDIR
 parameter 42
AC_REPORTCAST_VIEWTMPDIR_
 CLEANING_FREQUENCY parameter 43
AC_REPORTCAST_VIEWTMPDIR_FILE_
 PERSISTENCY parameter 43
AC_REPORTCAST_
 VIEWTMPDIR_MAX_FILE parameter 42
AC_REPORTCAST_
 VIEWTMPDIR_MAX_SIZE parameter 42
AC_REPORTCAST_
 VIEWTMPDIR_VIRTUAL parameter 42
AC_REPORTCAST_LOCALE
 parameter 30, 31
AC_REPORTCAST_CONNECTION_CACH
 E_SIZE parameter 39, 192
AC_REPORTCAST_CONNECTION_CACH
 E_TIMEOUT parameter 40, 192
AC_REPORTCAST_DEFAULT_REPORT_SE
 RVER parameter 23
AC_REPORTCAST_HOST parameter 25, 200
AC_REPORTCAST_LOGIN_TIMEOUT
 parameter 168
AC_REPORTCAST_PORT parameter 24, 200
AC_REPORTCAST_REQUEST_TIMEOUT
 parameter 40
AC_REPORTCAST_SECURITY_LIBRARY
 parameter 185, 186
AC_REPORTCAST_USE_LOGIN_FORM
 parameter 167
AC_SERVER_REPLY_SOCKET_BASE
 parameter 195
AC_SERVER_REPLY_SOCKET_COUNT
 parameter 195
AC_SERVER_RPC_SOCKET_BASE
 parameter 191
AC_SERVER_RPC_SOCKET_COUNT
 parameter 191
AC_SERVER_SOCKET_BASE parameter 26,
 27
AC_SERVER_SOCKET_COUNT
 parameter 27
AC_VIEW_SERVER_SOCKET_BASE
 setting 197
AC_VIEW_SERVER_SOCKET_COUNT
 setting 197
AC_WEBAGENT_AUTHENTICATION_PR
 OMPT setting 168
AC_WEBAGENT_UNAUTHORIZED_STAT
 US_CODE setting 169
access restrictions 33, 43, 164, 188, 203
access rights 104, 132, 133
 See also privileges
accessing
 custom templates 46
 e.Reporting Server 2, 169
 home folders 81
 Report Encyclopedia in web
 environments 5–8
 Report Encyclopedia items 16
 report files from non-default ports 26
 ReportCast 31, 32, 53
 ReportCast channels 7
 ReportCast templates 49–51
 reports in Windows environments 32
accessing online help xiii
__accessToGrant variable 104
.ACHTML files 46
ACSERVERALIAS property 149
__active directive 82, 111
active request
 displaying URL for 148
 getting 126

- Active template 61
- Activedetail template 61
- ActiveRequestsURL variable 148
- Actuate
 - home page xi
 - release notes xi
- Actuate Basic
 - syntax conventions xvii
- #Actuate keyword 119
- Actuate LRX x
 - data reply connections for 190
 - viewing reports 96, 198
- Actuate product summary viii
- Actuate Scripting Language
 - command reference 118–145
 - command separator 85
 - command types listed 118
 - commands listed 119
 - directive formats 78–80
 - directives described 80–115
 - directives listed 81–83
 - embedding in HTMLpages 46
 - getting number of processed
 - commands 146, 147
 - overview 118
 - overview to directives for 76
 - template mappings for 62
 - variables described 145–161
- acweb keyword 76, 79
- AcWebSecurityAuthenticate function 181
 - overview 180
- AcWebSecurityStart function 180
 - overview 180
- AcWebSecurityStop function 182
 - overview 180
- ad hoc parameters 159
- adding ReportCast channels 82
- AdHoc variable 159
- ?admin directive 35
- __?admin directive 81, 82, 111
- __admin keyword 79
- Admin template 61
- ADMIN.ACHTML 48
- administration page
 - accessing 61, 81
 - creating 48
 - getting URL for 146
 - implementing security from 166
- administration tasks
 - listing 48
 - managing ReportCast 20
 - managing ReportCast channels 34–38
 - setting user name and passwords 166
- administrative directives 35
- administrative groups 148
- administrative security 164
- administrator privileges 34
- AdminURL variable 146
- Advanced e.Reporting Server x
 - after keyword 120
- __ageDays variable 107
- __ageHours variable 107
- AgentVersion variable 146
- aging expiration policies 138, 160
- AIX systems
 - installing ReportCast 23
- Alias variable 158
- aliases 148, 158
 - for non-default server ports 26
- ANALYSIS format 92
- analysis tools (third-party) viii
- anonymous users 166
- Apache web servers 167, 168
- application programming interfaces viii
- __archiveBeforeDelete variable 107
- __archivePolicy variable 107
- ascending sort order 85, 123
- assigning roles to channels 38
- assigning values to variables 144
- asterisk (*) as wildcard 123
- authentication
 - bypassing 166
 - default 169, 170
 - enabling 181
 - enabling form-based login 167
 - failing 169
 - overriding 171, 175
 - overview 164
 - process described 170, 173
- authentication filters 184
- authoring tools 55
- automatic cleaning process 43
- automating reporting tasks ix

B

- base URLs 26, 33
- Basic Authentication header 171
- before keyword 120
- blank pages 31
- blank requester forms 99
- bookmarks 87
- border colors 70
- browser plug-ins 32
- browsers *See* web browsers
- browsing 192
- BundleRox variable 104
- bundling flag 104
- buttons 76
- Byfr report 100

C

- cache 146, 167
- caching connections 39, 40, 182, 189, 192
- Cancel template 61
- CANCEL.ACHTML 47
- cancelling requests 11, 47, 81, 108
- Cancelled Request Confirmation page 47
- CancelURL variable 154, 156
- cascading style sheets 90
- case sensitivity
 - file types 123
 - report parameters 104
 - ReportCast extensions 182
 - search results 96
 - variables 104
- case statements 119
- CGI component 22
- CGI programs
 - formatting directives for 78
- <cgi script location> parameter 79
- CGI scripts
 - accessing ReportCast with 31, 32
 - implementing security with 33
- changing
 - border colors 70
 - copyright information 71
 - file names 32
 - localemaps 31
 - passwords 12, 182
 - ReportCast channels 35, 38

- ReportCast Server 25
- ReportCast templates 56, 60
 - examples for 67, 69
- sort orders 123
- TCP/IP ports 24
- unauthorized access status 169
- user preferences 80
- channel administration page
 - creating 48
 - displaying 37, 113
- Channel Contents page 48
- channel creation form 37
- Channel Detail page 48
- channel details form 82
- Channel directive 80
- channel directives 111–113
- channel icons 7
- channel objects 52
- channel parameter variables 114
- Channel Subscription page 17, 48
- channel subscription variables 115
- channel URLs 113–115
- channel variables 160
- Channeladmin template 61
- CHANNELADMIN.ACHTML 48
- Channelcontents template 61
- CHANNELCONTENTS.ACHTML 48
- Channeldetail template 61
- CHANNELDETAIL.ACHTML 48
- Channelitemdetail template 61
- __ChannelName variable 114
- ChannelName variable 160
- __channels directives 35
- __channels variable 106
- channels x, 4
 - accessing 7
 - adding 82
 - administering 34–38, 112
 - assigning roles 38
 - cached connections for 189, 192
 - changing 35
 - changing attributes for 82
 - changing details for 38
 - checking contents 17
 - creating 35, 37–38
 - deleting 35, 38, 82
 - directives for 35

- channels (*continued*)
 - displaying contents 34, 48, 80, 113
 - displaying subscription forms 82
 - getting current subscriptions 135
 - getting granted permissions 132, 133
 - getting information 160
 - getting URLs 148, 160
 - listing available 112
 - listing contents 35
 - overview 34
 - removing items 38
 - specifying icons for 37
 - subscribing to 17, 114
 - testing current subscriptions 160
 - unsubscribing from 17
 - updating contents 38
- channels administration page 35, 36
- channels keyword 121
- channels subscription form 35
- channels variable 115
- Channelssubscription template 61
- CHANNELSUBSCRIPTION.ACHTML 48
- ChannelsURL variable 148
- ChannelURL variable 160
- character matching 123
- character sequences
 - for escape sequences 32
 - for file names 33
- character strings
 - assigning to variables 145
- cleaning process 43
- client validation 169
- colors
 - changing border 70
- comma separated values 92
- command separator 85
- CommandName variable 147
- commands 3
- commands *See* Actuate Scripting Language
- __Comment variable 114
- Comment variable 153, 160
- communications
 - for networked environments 189–200
 - for web servers 26, 78
 - for Windows platforms 24
 - processes for Internet 201, 203
- company names and logos
 - example for creating 69
- __completed directive 35, 82, 111
- completed requests
 - displaying URLs for 148
 - getting date/time 156
- CompletedRequestsURL variable 148
- CompletedTime variable 156
- componentID keyword 87, 90
- ComponentID variable 150, 151
- compressed DHTML formats 88
- concurrency 194, 198
- conditional expressions 142
- conditional statements 119
- <config> parameter 28, 79
- configurations
 - cached connections 192
 - customizing 28
 - data reply ports 194
 - dual firewalls 202–206
 - e.Reporting Server ports 190–200
 - mapping to locales 29–31
 - network ports 189
 - ReportCast ports 26–28
 - secured networks 201
 - security extensions 185
 - setting directory paths for 29
 - split ReportCast 198–200, 206
 - view process ports 197
 - viewing options 42
- configuration-specific templates 50
- Confirm template 61
- CONFIRM.ACHTML 47
- confirmation forms 47, 48
- Confirmation page 48
- connections 193
 - caching 39, 40, 182, 189, 192
 - getting 136
 - IP addresses and Internet 189, 201, 203, 205
 - securing 189
 - setting for e.Reporting Server 190
 - setting number of open 39
 - setting port numbers for 26–28
 - setting registered ports for 190, 191
 - setting Report Encyclopedia 192, 198
 - setting timeout intervals for 39
 - user load and 191

- content keyword 120
- contents page
 - creating for folders 8
 - displaying for DHTML reports 90
 - displaying reports from 88
 - getting 83
 - getting information about 150
 - listing DHTML items in 138
 - setting up 48
 - testing for sub-items 150
- ControlName variable 150
- cookies 31
- copying
 - templates 29
- copyright information
 - changing 71
 - creating 49
- corporate networks 188, 203
- Created variable 153
- creating
 - custom copyright information 71
 - custom requester forms 59, 100, 127
 - customized company logos 69
 - ReportCast channels 35, 37–38
 - ReportCast templates 53, 55, 67
 - web pages 46
- creating report-specific online help xv
- criteria keyword 122
- CSV format 92
- current date 146
- current folder variables 152
- current request variables 154
- current time 146
- CurrentPage variable 149
- CurrentPages variable 156
- CurrentSize variable 156
- CurrentTOCParentID variable 149
- custom configurations 29–31
 - creating 28
 - setting directory paths for 29
- custom default forms 46
- Custom directory 29
- custom directory 55
 - adding templates to 49
- Custom Folder page 47
- custom Requester forms
 - attaching to reports 59
 - creating 59, 100, 127

- examples of 100
 - overview 99
- custom templates
 - accessibility 46
 - adding to Encyclopedia 63, 64
 - creating 55, 58, 67
 - examples 67–71
 - naming 61
 - overview 46
 - setting directory paths 29
- custom web pages 63, 64

D

- data
 - getting for specific component 82
 - getting formatted 82
- data packet transmissions 40
- data protection 188–189
- data reply connections 190
- data reply ports 194
- data reply requests 195
- databases ix
- date formats 104
- DateAndTime variable 158
- dates
 - checking for web pages 6
 - getting 146, 156
 - getting for status information 158
 - last modification 152, 154
 - printing 29
 - setting default 154
- __dateToDelete variable 107
- __default keyword 79
- default configuration 28
- default directories 53
- default file names 32
- default forms 46
- default port allocation 191
- default reporting server 23–24
- default reporting web site 2
- default settings (system-wide) 149
- default templates 3, 46, 58
- default timeout interval 6
- default user authentication 164, 169, 170
 - overriding 171, 175
- Default variable 158
- DefaultDate variable 154
- DefaultTime variable 154

- ?DeleteStatus directive 81, 109
- DeleteURL variable 156
- deleting
 - channel items 38
 - ReportCast channels 35, 38, 82
 - status information 81, 109, 156
- demand paging 97
- DemandCachePageSize registry key 40, 41
- dependencies
 - ReportCast template files 49, 60
- deploying online reports
 - data protection strategies for 188–189
 - overview 188, 200
 - setting up ReportCast for 4
- desc keyword 123
- descending sort order 85, 123
- Description variable 158
- destination ports 195
- ?Detail directive 81, 110
- details form 35
- DetailURL variable 156
- DHTML reports 2
 - configuring ports for 190
 - customizing searches for 73
 - displaying 12, 48, 82, 86–91, 196
 - displaying contents page 90
 - displaying navigation bars for 90
 - expiring 87
 - formatting 88
 - generating navigation bar for 86
 - getting available versions 83, 124
 - getting content for 87
 - getting information on contents 150
 - getting information on current views 149
 - getting style sheets for 90
 - listing items in contents page 138
 - sample application for 16
 - scaling 87
 - searching 12–16, 81, 83, 91, 151
 - supporting locales for 31
- ?Direction directive 81
 - issuing 84–86
- directives *See* Actuate Scripting Language
- directories
 - displaying full path names 152, 154, 155
 - native file systems 42
 - restricted access and temporary 43
 - returning paths for 136, 152
 - securing 43
 - setting paths for custom configurations to 29
 - setting up templates 53
- DISPLAY format 92
- display options 11, 42
- displaying
 - available channels 112
 - channel contents 34, 48, 80
 - current users 147
 - DHTML reports 12, 48, 82, 86–91, 196
 - Encyclopedia contents 16, 80, 81, 125
 - factory requests 82
 - folder contents 8, 16, 47, 84, 125
 - folder lists 125
 - HTML source files 64
 - information for requests 47
 - navigation bar 90
 - online documentation xiii
 - PDF files 196
 - report files 86
 - reports 2, 16, 81, 96
 - reports in native file systems 41
 - reports in secured environments 193, 196
 - reports in specific formats 82
 - reports with Actuate LRX 198
 - request details 110
 - status information 47, 109
 - Table of Contents 90
 - URLs 77
- DLLs 20, 175
- document files
 - getting path links for 136
- Document Source command 64
- documentation x, xii
 - online xiii
 - syntax conventions xvii
 - typographical conventions xvi
- Download/Print link 88
- downloading
 - entire documents 81, 96
 - report files 80, 96, 98
- downloading Actuate Viewer LRX 12
- DownloadURL variable 155
- ?drop directive 35, 81, 82, 112
 - usage example 108

- dual firewalls 202–206
 - with proxy servers 205–206
- dynamic HTML *See* DHTML
- dynamic-link libraries 20

E

- e.Analysis application ix
 - customizing 71, 72
 - launching 72
 - overview 16
- e.Report Designer ix
- e.Report Designer Professional viii
- e.Reporting Server x
 - accessing 2, 169
 - caution for setting up 206
 - changing locales for 31
 - configuring viewing options 42
 - connecting to 26–28, 190
 - converting reports 31
 - downloading files 96, 98
 - getting connections 136
 - getting current state 131, 157
 - getting information about 148
 - logging onto 166–167
 - masking URLs for 33
 - overview xi
 - receiving data reply requests 195
 - registering ports 191
 - removing channels from 35
 - reporting error conditions for 161
 - securing for networked environments 188, 189
 - sending e-mail notifications 39
 - setting default 23–24
 - setting number of open connections for 39
 - setting template dependencies for 49
 - specifying custom configurations for 28
- e.Reporting Server objects 51
- e.Reporting Server port
 - configuring 190–200
 - overview 189
- e.Reporting Server variables 148
- e.Reporting Suite vii
 - product summary viii
- editing tools 55
- elapsed time 146, 156
- ElapsedTime variable 146, 156
- ellipsis (...) as wildcard 123
- e-mail notifications *See* notifications
- end list keyword 120
- End User Desktop ix
- EndHitNumber variable 151
- Entries variable 154
- enu directory 54
- environment variables 165
- error codes and numbers 161
- error messages 161
- error objects 52
- Error page 48
- Error template 62
- error variables 161
- ERROR.ACHTML 48, 169
- ErrorClass variable 161
- ErrorNumber variable 161
- ErrorText variable 161
- escape sequences 32
- __exclude variable 106
- executable files
 - attaching Requester forms to 59
 - getting full paths 155
 - getting information about 154
 - getting version 156
 - running 9, 49
- executing report executables 9, 49
- executing reports 98
 - creating completed notices for 134
 - creating user-specified requests for 127, 129
 - getting schedules for 134
 - predefined forms for 47
 - process for 194
 - requests for 6
- Expandable variable 150
- __Expiration variable 114
- expiration policies 138, 160
- Expiration variable 160
- Expirationpolicies keyword 122
- expressions 142
- ?extractSearchResults directive 83, 96

F

- F1 key xiii
- Factory directives 80, 98–102

- factory requests
 - displaying active 82
 - displaying completed 82
 - getting information about 111
 - getting scheduled 82
- file creation dates/time 153
- file dependencies
 - ReportCast templates 49, 60
- File Detail page 48
- file directives 80, 96–98
- file lists 122
 - sorting items 122, 123
- file names
 - custom ReportCast templates 61
 - default report 32, 42
 - getting extension 153, 155
- file types
 - described 99
 - getting 153
 - identifying 32
 - listing 123
- File Versions page 48
- FileDetail template 62
- FILEDETAIL.ACHTML 48
- ?Files directive 81
- files
 - See also* output files; template files
 - accessing report 26
 - attaching Requester forms to 59
 - creating ReportCast template 55
 - displaying available versions 48
 - downloading 96, 98
 - downloading report 80, 81
 - generating from conditions 142
 - generating PDF 88
 - generating report 8–11, 80, 98
 - generation request forms for 127, 129
 - getting aging expiration 138
 - getting available versions 81, 83, 124
 - getting information 153
 - getting path links for 136
 - getting size 153
 - listing for specific folders 81
 - listing properties 48
 - localemap 31
 - renaming template 57
 - report types 99
 - running report executable 9
 - running report executables 49
 - securing temporary 43
 - viewing in folders 86
 - viewing PDF and XML 196
- files keyword 121
- FileSize variable 153
- FileType variable 153
- FileVersion variable 153
- FileVersionName variable 153
- Fileversions template 62
- FILEVERSIONS.ACTML 48
- finding data
 - in DHTML reports 12–16, 81, 83, 91, 151
- finding template files 50, 57
- firewalls
 - configuring dual 202–206
 - defined 188
 - setting up 201
 - specific ports for 26
 - split ReportCast configurations for 198
- folder contents pages 8
- folder directives 80, 84–86
- folder item objects 51
- folder item variables 152
- Folder Listing page 47
- folder objects 51
- folder variables 152, 154
- FolderHeader variable 152
- Folderlist template 62
- FOLDERLIST.ACHTML 47
- FolderName variable 152, 154
- FolderPath variable 152
- folders
 - accessing home 81
 - displaying contents 8, 16, 47, 84, 125
 - displaying current 125
 - displaying report files only 86
 - displaying URLs for 148
 - getting file lists for 122
 - getting information 154
 - getting information on current 152
 - getting information on items 152
 - getting paths for 136, 152
 - listing contents 81

- folders (*continued*)
 - modification dates/time 152
 - setting templates dependencies for 49
 - sorting items 85, 122, 125
- folders keyword 121
- FolderTitle variable 152
- forced reloads 7, 167
- format keyword 88, 90
- format preferences 149
- formatting report layouts 88, 90
- formatting ReportCast directives 78–80
- formatting search results 92
- forms
 - customizing Requester 59
 - displaying subscription 82
 - login authentication 167
 - predefined 46, 47
 - ReportCast channel attribute 82
 - setting dependencies for 49, 60
 - setting up Requester 46
 - subscribe/unsubscribe 48
- frames 48
- functions 180

G

- generating HTML pages 2
- generating report files 8–11
 - creating completed notices for 134
 - creating user-specified requests for 127, 129
 - directives for 80, 98
 - getting schedules for 134
 - with conditions 142
- generating web pages 118, 145
- ?GetDynamicData directive 82, 87
- ?GetReportData directive 82, 89
- ?GetStyleSheet directive 83, 90
- globally unique IDs 153, 155
- Granted variable 157
- grants keyword 121
- graphical user interfaces 20
- graphics files 37
- GroupName variable 158
- __groups variable 106
- groups
 - generating lists for 131, 134, 157
 - getting admin status for 148
 - getting information 156
 - getting report parameter 158
- GUIDs 153, 155
- GUIs 20

H

- headers 49
- Headline variable 104, 156
- help xiii–xv, xv–xvi
 - creating report-specific xv
 - usage overview xiii
- help directories 55
- Help menu xiii
- help topics xiii
- helper applications 32
- Helpers command 32
- hexadecimal code 32
- hidden parameters 159
- Hidden variable 159
- Hide Search URL option 15
- HideText variable 159
- hierarchical paths 152
- __home directive 81
- home folder 81, 148
- HomeFolderURL variable 148
- Host variable 146
- hosts 79, 146, 147, 200
- HTML 46
- HTML authoring tools 55
- HTML pages
 - creating 46
 - generating 2, 118, 145
 - generating from conditions 142
 - masking URLs for 33
 - predefined 47
 - ReportCast directives and 76
 - viewing 97
- HTML reports 2
- HTML requester forms
 - See also* Requester forms
 - attaching to reports 59
 - customizing 59, 99, 127
 - examples of 100
 - filling out 10
 - generating 81, 98
 - selecting 60
 - setting up 46

- HTML scripting language *See* Actuate Scripting Language
- HTML source files 64
- HTML tags 55
 - embedding scripting language commands 46
- HTML template files
 - caching 146
 - creating 55
 - defining variables in 145
 - getting names 147
 - listed 47
 - loading 143
 - naming 61
 - overview 46
 - reloading 60, 61
- HTML templates 3, 77
 - accessibility 49–51
 - adding to Encyclopedia 63, 64
 - associating objects with 53
 - changing 56, 60
 - examples for 67, 69
 - customizing
 - examples 67–71
 - elements described 55
 - overview 46, 49
 - referencing subtemplates in 57
 - search paths 50, 58
 - searching for 57–59
 - setting directory structures 53
- HTTP communications protocol 78
- HTTP headers 182
- http parameter 78
- HTTP protocol 192, 199, 201
- HTTP unauthorized status codes 169
- HTTP_AC_REPORTCAST_BASE_URL
 - parameter 33
- HTTP_AC_REPORTCAST_PARSED_HEADE
 - RS parameter 32
- https parameter 78
- HTTPS protocol 192, 199, 201
- HTTPS support 164
- hyperlinks
 - as navigation tools 4
 - associated with ReportCast channels 17
 - associated with ReportCast directives 76
 - displaying 77

- for DHTML searches 12
 - getting associated directive 76
 - masking URLs for 33
 - specifying server ports for 26
- hypertext markup language *See* HTML
- HyperText Transfer Protocol *See* HTTP

I

- IBM AIX. *See* AIX systems
- icons 37
 - getting URLs for 160
- idle connections 40
- idle user session timeouts 168
- If command 119, 142
- ifempty keyword 120
- images 37
- images directory 55
- inbound ports 189
- Include command 119, 143
- Index template 62
- INDEX.ACHTML 47
- information directives 110–111
- initialization 180
- InputObject constant 144
- installation defaults 53, 58
- installing ReportCast for UNIX systems 23
- installing online documentation xiii
- internationalization 55
- Internet connections 201, 203
- Internet deployments
 - data protection strategies for 188–189
 - overview 200
- Internet Explorer
 - displaying links 77
 - reloading web pages 167
- Internet Information Server (IIS) 184
- IP addresses
 - displaying dotted-decimal 147
 - Internet connections and 189
 - Internet routings for 201, 203, 205
 - two-way mapping for 200
- IsAdministrator variable 148
- ISAPI filters 185
- IsBundledOutput variable 153
- IsExecutable variable 153
- ItemBaseName variable 152
- ItemName variable 152

ItemPathName variable 153
items keyword 121
iterating through lists 119
iteration scripting commands 118

J

JavaScript 46
jpn directory 54

K

keywords 121
See also specific

L

language groups 29
language support 55
large icons 160
 __LargeIconURL variable 114
LargeIconURL variable 160
LastModified variable 152, 154
LastModifiedBy variable 152
LatestURL variable 153
libraries 175
library functions 180
 __limit variable 106
 __limitNumber variable 106
LinkedFolderPath variable 152, 153
links
 as navigation tools 4
 associated with ReportCast channels 17
 associated with ReportCast directives 76
 displaying 77
 for DHTML searches 12
 getting associated directive 76
 masking URLs for 33
 specifying server ports for 26
List channels command 135
List command 119–122
 keywords listed 121
 overview 119
List criteria command 138
List expirationpolicies command 138
List files command 122–124
 specifying file type 123
List folders command 125
List grants command 132

List items command 125
list keyword 120
List members command 131
List notifications command 134
List parameters command 127
List paramgroups command 127
List pathlinks command 136
List printerprops command 135
List privileges command 133
List reportservers command 136
List requestforms command 126
List requests command 126
List resultfields command 139
List resulthits command 139
List schedules command 134
list statements 143
List status command 130
List tocitems command 138
List values command 129
List versions command 124
lists
 available file versions 124
 channel subscriptions 135
 current users 131, 134, 157
 defining 120
 DHTML contents 138
 directory paths 136
 folder contents 122
 folders 125
 iterating through 119
 object expiration 138
 parameter groups 128
 permissions status 132, 133
 printer properties 135
 report parameters 127, 129
 request schedules 134
 request status 130
 Requester forms 126
 requests 126, 155
 search criteria 139
 search hits 139
 server state 131, 157
Live Report Extension *See* LRX
loading reports in web browsers 34
__local keyword 79
local file systems 32
localemap file 31

- locales 54
 - custom configurations for 29–31
- localization settings 29
- locating data
 - in DHTML reports 12–16, 81, 83, 91, 151
- locating template files 50, 57
- login authentication 167
- Login dialog box 5
 - anonymous users 166
 - customizing 168
 - specifying unauthorized status for 164
- login forms 167
- login scripts
 - anonymous users 166
 - failing 171, 174
- LOGIN.ACHTML 167
- logins 164
- logos
 - example for creating 69
- logout forms 167
- LOGOUT.ACHTML 167
- loops 119
- LRX
 - data reply connections for 190
 - enabling 12
 - viewing reports 96, 198
- LRX (Live Report Extension) x

M

- Manuals directory xiii
- mapping locales 29–31
- masking URLs 33
- masks 159
- member objects 51
- member variables 157
- MemberName variable 157
- members keyword 121
- MemberType variable 157
- MIME types 32
- mode keyword 89
- modification dates/time
 - folders 152
 - report executables 154
- moving through Report Encyclopedia 4, 16
- multiple sort fields 124
- My Headlines tab 7

N

- Name variable 158, 159
- naming custom template files 61
- naming report files 32, 42
- NAT Static Address Translation
 - algorithm 204
- native file systems 41–43
- navigating through Report Encyclopedia 4, 16
- navigation bar
 - displaying 83, 90
 - generating 86
 - searching from 96
 - viewing DHTML pages from 88
- navigation buttons 97
- Navigation page 48
- Navigator (Actuate) 4
- Netscape Navigator
 - displaying links 77
 - implementing security for 184
 - reloading web pages 167
- Network Address Translation (NAT) 189, 200, 204, 206
- network communications protocol 189–200
- network security
 - configuration options 201
 - data protection strategies 188
 - overview 188
- new channel creation page 37
- new channel form 82
- New Channel option 37
- NewItem variable 153
- NextPollTime variable 160
- NextStartTime variable 157
- non-Actuate objects 41
- non-default web server ports 26
- notification groups 106
- notification lists 131, 134, 157
- notifications
 - configuring data reply ports 195
 - data reply connections for 190
 - overview 34
 - sending 39
- notifications keyword 121
- Now variable 146
- nph-actuate.cgi 32

- NSAPI extension 22
- NumberOfCommands variable 146
- NumberOfHitsOnPage variable 151
- numbers 29
- NumVersions variable 153

O

- object IDs 153
- object lists 122
 - sorting items 122, 123
- ObjectID variable 153
- objects
 - accessibility 53
 - associating with ReportCast templates 53
 - generating information for 81
 - getting detail information 110
 - getting expiration policies for 138
 - getting granted permissions for 132
 - getting URLs 147
 - initializing values for 144
 - ReportCast described 51
- ObjectURL variable 147
- on demand paging 97
- __onceDate variable 104
- __onceTime variable 105
- online documentation xii, xiii
 - syntax conventions xvi
 - typographical conventions xvi
- online help xiii–xv, xv–xvi
 - creating report-specific xv
 - usage overview xiii
- online reports vii
 - See also* reports
 - deploying 4, 188, 200
 - getting current 6
 - getting specific version 98
 - overview 8
- open connections 193
- open server x
- operation keyword 89
- outbound ports 189
- output 88
- output files
 - getting location 154, 155
 - getting number of pages 157
 - getting size 156, 157
 - getting version 156

- OutputLocation variable 155
- __outputname variable 106
- OutputName variable 155
- OutputObject constant 144
- OutputPathName variable 155
- OutputVersion variable 156
- OutputVersionName variable 156
- __overwrite variable 105
- OwnerName variable 153
- owners 153

P

- packets 40
- page headers 49
- page keyword 87, 90
- page ranges 157
- Pages variable 157
- parameter group objects 51
- parameter group variables 158
- parameter groups
 - displaying 158
 - generating lists of 128
- parameter objects 51
- parameter values files 157
- parameter variables 104, 114, 158
- parameters 98
 - data reply connections for 190
 - defining in directives 104
 - displaying default 158
 - generating requester forms for 99
 - getting available 127
 - getting information 158
 - masking values 159
 - setting 129
 - specifying default 127
- parameters keyword 121
- ParamGroups keyword 121
- Parent URL variable 152
- ParentFolderPath variable 152
- parsed-header CGI scripts 31, 32
- parsed-header output 32
- passwords
 - anonymous users 166
 - changing 12, 182
 - setting 166
- <path> parameter 80

- path names
 - current folder 136, 152
 - executable files 155
 - folder items 152
 - output files 155
 - report files 154
- PATH_INFO variable 33
- pathlink keyword 122
- PathName variable 152
- .PDF files xiii
- PDF files xi
 - generating 88
 - supporting locales for 31
 - viewing 190, 196
- permissions 104, 132, 133
 - See also* privileges
- Personalchcontents template 62
- ph-actuate.cgi 32
- PlainComment variable 160
- plug-ins 32
- __PollingInterval variable 114
- PollingInterval variable 160
- port mapper
 - connections to 193
 - overview 190
 - registering ports with 191
- ports
 - changing TCP/IP 24
 - configuring data reply 194
 - configuring for e.Reporting Server 190–200
 - configuring for networks 189
 - configuring for ReportCast 26–28
 - configuring view process 197
 - dual firewalls and 203
 - inbound defined 189
 - Internet connections for 201, 203
 - outbound defined 189
 - registering 190, 191
 - resetting 26
 - searching for available 191
 - using non-default server 26
- predefined templates 3, 46
- predefined web pages 47
- preferences
 - changing 80
 - getting 149
 - setting 11, 48
- PreviousURL variable 147
- printed documentation x, xii
 - syntax conventions xvii
 - typographical conventions xvi
- Printer property objects 52
- printer variables 159
- PrinterProps keyword 121
- printers
 - generating property lists 135
 - getting information for current 159
- printing numbers and dates 29
- printing reports 88, 157
- PrintRange variable 157
- __priority variable 105
- Priority variable 155
- __priorityValue variable 105
- privilege variables 157
- PrivilegeName variable 157
- privileges 34, 157
- privileges keyword 121
- process groups 156
- ProcessGroup variable 156
- product summary viii
- product updates xi
- programming interfaces viii
- property lists 135
- protection *See* security
- providing report-specific online help xv
- proxy servers 188, 189
 - dual firewall configurations for 205–206
- publish and subscribe 34
- push distribution 34
- push technology 4, 34

Q

- QUERY_STRING variable 33
- question mark (?) as wildcard 123

R

- RawErrorText variable 161
- RawPollingInterval variable 160
- RCASE *See* ReportCast Agent Security Extension

- RCSE functions 180
- RCSE *See* ReportCast Security Extension
- reconstructing URLs 33
- recurring schedules 105
- __recurringTime variable 105
- redirecting web browsers 26
- Refresh Search URL option 15
- refreshes 160
- REG_SZ data type 30
- Regedit 23, 164
- registering network ports 190, 191
- registering plug-ins 32
- registry keys 164
- release notes xi
- reloading templates 60
- reloading web pages 7, 167
- RemoteAddress variable 147
- RemoteHost variable 147
- renaming
 - report files 32
 - template files 57
- report definition objects 51
- report definition variables 154
- report definitions
 - getting 154
- Report Encyclopedia
 - accessing in web environments 5–8
 - adding custom templates 63, 64
 - connecting to 192, 198
 - displaying contents 16, 80, 81, 125
 - displaying folder contents 8, 16, 47, 84, 125
 - displaying report files only 86
 - enabling MIME types for 32
 - generating information for objects 81
 - getting detail information 110
 - getting granted permissions 132, 133
 - getting item expiration lists 138
 - items containing spaces or special characters 33
 - listing items 122
 - logging into 5
 - navigating through 4, 16
 - securing in native file systems 43
 - sorting folder contents 85, 122, 125
 - storing non-Actuate objects in 41
 - updating items 7
- Report Encyclopedia web page 5, 6
- report files
 - accessing 26
 - default names for 32
 - displaying available versions 48
 - downloading 80, 81, 96, 98
 - generating 8–11, 80, 98
 - generating from conditions 142
 - getting aging expiration 138
 - getting available versions 81, 124
 - getting information 153
 - getting size 153
 - naming for native file systems 42
 - types listed 99
 - viewing 86
- report generation requests
 - canceling 11
 - creating completed notifications for 134
 - generating schedules for 134
 - generating status information for 81
 - getting status 108
 - setting conditions for 142
 - submitting 10, 81, 98, 102, 103
 - user-specified 127, 129
- report object protocol 39
- report server API ix
- <report URL> directive 81
- report viewing options 11–12
- Report Viewing Options link 11
- Report Viewing Options page 48
- ReportCast
 - accessing with CGI scripts 31, 32
 - changing TCP/IP ports for 24
 - configuring cached connections for 192
 - configuring port numbers for 26–28
 - connecting to port mapper 193
 - customizing configurations for 28
 - getting information about 146
 - implementing security 164–166
 - installing on UNIX systems 23
 - mapping locales for 29–31
 - overview 2, 20
 - search processes for 57
 - setting active port for 26
 - setting default server for 23, 24
 - setting e.Reporting Server ports for 190, 191

- ReportCast (*continued*)
 - setting request timeouts for 40
 - setting up 4
 - starting and stopping 20–23
 - tuning performance for 39–41
 - UNIX components for 22
 - viewing options 11–12, 48
- ReportCast Agent
 - accessing Encyclopedia 193
 - dual firewall configurations for 202–206
 - overview 2, 20
 - receiving requests 200
 - setting TCP/IP ports for 24
 - setting up native files systems for 41
 - starting and stopping 21
- ReportCast Agent Security Extension (RCASE) 164, 169, 170, 185
- ReportCast channel directives 111–113
- ReportCast channel objects 52
- ReportCast channel URLs 113–115
- ReportCast channel variables 160
- ReportCast channels x, 4
 - accessing 7
 - adding 82
 - administering 34–38, 112
 - assigning roles 38
 - cached connections for 189, 192
 - changing 35
 - changing attributes for 82
 - changing details for 38
 - checking contents 17
 - creating 35, 37–38
 - deleting 35, 38, 82
 - directives for 35
 - displaying contents 34, 48, 80, 113
 - displaying subscription forms 82
 - getting current subscriptions 135
 - getting granted permissions 132, 133
 - getting information 160
 - getting URLs 148, 160
 - listing available 112
 - listing contents 35
 - overview 34
 - removing items 38
 - specifying icons for 37
 - subscribing to 17, 114
 - testing current subscriptions 160
 - unsubscribing from 17
 - updating contents 38
- ReportCast commands 118–145
 - listed 119
 - overview 118
 - syntax 119
 - types listed 118
- ReportCast directives
 - examples 115
 - formatting 78–80
 - getting associated link 76
 - issuing 76
 - listed 81–83
 - mapping to templates 62
 - overview 76, 77
 - syntax 83
 - types described 80–115
- ReportCast objects 51–53
 - accessibility 53
 - getting detail information 110
 - relationships defined 52
- ReportCast scripting language *See* Actuate Scripting Language
- ReportCast Security Extension (RCSE)
 - administrative tasks 166
 - changing unauthorized access status 169
 - customizing Login dialogs for 168
 - development guidelines 182–185
 - enabling 164–165, 185
 - overriding default authentication 171, 175
 - overview 164, 169
 - testing 182
- ReportCast Server
 - accessing Encyclopedia 193
 - changing 25
 - dual firewall configurations for 203
 - overview 2, 20
 - setting TCP/IP ports for 24
 - split ReportCast configurations for 198, 200
 - starting and stopping for UNIX platforms 22
 - starting and stopping for Windows platforms 21
- ReportCast startup script 22, 24

- ReportCast template files
 - caching 146
 - creating 55
 - defining variables in 145
 - getting names 147
 - listed 47
 - loading 143
 - naming 61
 - overview 46
 - reloading 60, 61
- ReportCast templates 77
 - accessibility 49–51
 - adding to Encyclopedia 63, 64
 - associating objects with 53
 - changing 56, 60
 - examples for 67, 69
 - customizing 55, 58, 67
 - examples 67–71
 - default 3, 46, 58
 - elements described 55
 - mapping directives to 62
 - overview 46, 49
 - referencing subtemplates in 57
 - search paths 50, 58
 - searching for 57–59
 - setting directory structures 53
- ReportCast variables 146
- ReportDefinition constant 144
- report-generation requests *See* Requester forms; requests
- Reporting Server
 - accessing 2, 169
 - caution for setting up 206
 - changing locales for 31
 - configuring viewing options 42
 - connecting to 26–28, 190
 - converting reports 31
 - downloading files 96, 98
 - getting connections 136
 - getting current state 131, 157
 - getting information about 148
 - logging onto 166–167
 - masking URLs for 33
 - overview xi
 - receiving data reply requests 195
 - registering ports 191
 - removing channels from 35
 - reporting error conditions for 161
 - securing for networked environments 188, 189
 - sending e-mail notifications 39
 - setting default 23–24
 - setting number of open connections for 39
 - setting template dependencies for 49
 - specifying custom configurations for 28
- Reporting Server objects 51
- Reporting Server variables 148
- reporting sites 2
 - getting current report 6
- reporting solutions vii
- ReportName variable 154, 155
- ReportOutputPath variable 154
- ReportPath variable 154
- ReportPathName variable 155
- ?Reports directive 81
- reports vii
 - accessing 32
 - creating online help for xv
 - deploying over web environments 4, 188, 200
 - display options 11, 42
 - displaying in native file systems 41
 - displaying in secured environments 193, 196
 - displaying in specific formats 82
 - displaying in web environments 2, 16, 81, 96
 - getting available 34
 - getting current 6
 - getting information about 154
 - getting specific page 82
 - getting specific version 98
 - loading 34
 - overview 8
 - restricting access to 33, 43, 188
 - running 98, 194
 - requests for 6
 - specifying location of 39
 - tuning viewing performance 40–41
 - viewing with Actuate LRX 198
- <reportserver> parameter 79
- ReportServerName variable 146
- ReportServers keyword 122
- ReportURL variable 154, 156

- ReportVersion variable 156
- ReportVersionName variable 156
- reqsrvr.sh 28
- ?Request directive 81, 98
- Request Confirmation page 47
- request confirmation page 129
- Request Detail page 47
- Request folder objects 51
- Request Folder page 47
- request folders 154
- request headlines 156
- request IDs 155, 156
- request item variables 155
- request objects 51
- request parameters 98
- request status page 40, 47, 130
- request status variables 158
- Request template 62
- REQUEST.ACHTML 47
- Requestdetail template 62
- REQUESTDETAIL.ACHTML 47
- RequestedBy variable 155
- Requester API viii
- Requester forms
 - attaching to reports 59
 - customizing 59, 99, 127
 - examples of 100
 - filling out 10
 - generating 81, 98
 - listing available 126
 - overview 34
 - predefined 47
 - selecting 60
 - setting request timeouts 40
 - setting up 46
- Requester page 9, 47
- Requestermenu template 62
- REQUESTMENU.ACHTML 47
- RequestForms keyword 121
- RequestID variable 155
- Requestlist template 62
- REQUESTLIST.ACHTML 47
- requests 34
 - canceling 11, 47, 81, 108
 - concurrent data 195
 - deleting 109
 - displaying active factory 82
 - displaying completed factory 82
 - displaying information about 47
 - entering DHTML search 12
 - executing 102
 - generating notification lists for 131, 134, 157
 - generating server-specific 126
 - generating status information for 80, 81
 - getting elapsed time 146, 156
 - getting information 155
 - getting information about factory 111
 - getting number 154
 - getting priority 155
 - getting scheduled date/time 157
 - getting scheduled factory 82
 - getting schedules 134, 159
 - getting status 11, 108, 155, 158
 - getting type 155
 - sending 6
 - sending to ReportCast Agent 200
 - setting timeout intervals for 40
 - submitting for DHTML searches 83, 96
 - submitting for report generation 10, 81, 98, 102, 103
 - timing out 6
 - user-specified 127, 129
 - viewing details about 110
- requests keyword 121
- ?requestSearch directive 83, 91
- requestSearch directive 95
- RequestStatus variable 155
- RequestType variable 155
- RequestURL variable 155, 156
- __requestUserPreferences directive 82
- required parameters 159
- Required variable 159
- resetting ports 26
- resetting unauthorized status 169
- <resource> parameter 80
- response headers 31, 32
- response pages (predefined) 47
- restricting access to reports 33, 43, 188
- restricting data access 164, 188, 203
- Resultfields keyword 122
- Resulthits keyword 122
- ?retrieveCreation directive 35, 82, 112
- ?retrieveDetails directive 35, 82, 112

- ?retrieveForm directive 35, 82, 111
- __Roles variable 114
- roles
 - assigning to channels 38
 - getting current 131, 157
 - getting granted permissions 132, 133
- root folder 148
- RootFolderURL variable 148
- ROTP syntax 39
- routers 205
- RPC protocol 191
- running report executables 9, 49
- running reports 98
 - creating completed notices for 134
 - creating user-specified requests for 127, 129
 - getting schedules for 134
 - predefined forms for 47
 - process for 194
 - requests for 6
- run-time access 145

S

- sample application ix, 16
- satellite dish icon 37
- Save To Volume dialog box 59
- scalingFactor keyword 87, 89
- schedule objects 51
- schedule variables 159
- __scheduled directive 82, 111
- scheduled requests
 - displaying URLs for 148
 - getting start date/time 157
- Scheduled template 62
- Scheduleddetail template 62
- ScheduleDescription variable 159
- ScheduledRequestsURL variable 148
- __schedulePeriod variable 105
- schedules
 - displaying 159
 - listing 134
- schedules keyword 121
- __scheduleType variable 104
- scope
 - ReportCast templates 49
- scripting language
 - command reference 118–145
 - command separator 85
 - command types listed 118
 - commands listed 119
 - directive formats 78–80
 - directives described 80–115
 - directives listed 81–83
 - embedding in HTMLpages 46
 - getting number of processed commands 146, 147
 - overview 118
 - overview to directives for 76
 - template mappings for 62
 - variables described 145–161
- scripts
 - accessing ReportCast with CGI 31, 32
 - adding locale mappings to UNIX 31
 - providing security with CGI 33
 - ReportCast service startup 22, 24
- search criteria 96
 - customizing 93, 94
 - getting information about 150
 - getting list of 139
 - removing 14
 - specifying 13, 92, 95
- search criterion variables 150
- search directives 81, 91
- search extension API viii
- search hits
 - getting information on 151
 - returning 139
- Search link 12
- search paths 50, 58
- search processes 57
- Search Request page
 - customizing search criteria for 93, 94
 - displaying 12, 83, 91
 - entering search criteria 13, 92
 - removing search criteria 14
- search requests
 - submitting 83, 96
- search result field variables 151
- search result hit variables 151
- search result set variables 151
- search results 88, 92
- search URL directive 15
- searches (help topics) xiv

- searching DHTML reports 12–16, 81, 83, 91, 151
- searching template files 50, 57
- ?searchReport directive 91
- searchReport directive 91, 93, 95
- SearchValue variable 150
- secure network environments 188, 189
- Secure Sockets Layers 78, 164
- secured servers 164
- security
 - administrative tasks 166
 - changing unauthorized access status 169
 - configuring 185
 - customizing Login dialogs for 168
 - data protection strategies for 188–189
 - guidelines for developing 182–185
 - implementing ReportCast 164–166
 - implementing with CGI scripts 33
 - native file systems 43
 - overview 163, 169, 187
 - server-specific information 184
 - testing 182
- security library
 - closing 182
 - creating 175
 - declaring functions in 182
 - example code 183
 - initializing 180
 - loading 180
- security network configuration options 201
- security variables 164–165
 - setting for UNIX systems 186
 - setting for Windows NT 164, 186
- select modifier 93
- Selected variable 150
- selection scripting commands 119
- select-only criteria 93
- sending e-mail notifications 39
- sending ReportCast directives 76
- sending requests 6, 200
- separator character 85
- sequential scripting commands 118
- server objects 51
- Server variable 148, 161
- ServerAlias variable 148
- ServerFeatureLevel variable 148
- servers x, 2
- changing ReportCast 25
- communications protocol for 78
- configuring network ports 189
- configuring specific ports for 26, 27, 28
- connecting to e.Reporting 190
- default authentication for 167
- defining configuration-specific templates 50
- getting active requests 126
- getting connections 136
- getting current state 157
- getting information about 146, 147, 148
- getting system-wide defaults 149
- logging onto 166–167
- masking URLs for 33
- native files systems for 41–43
- overview to e.Reporting xi
- receiving data reply requests 195
- registering ports for 191
- reporting error conditions for 161
- resetting port numbers 26
- response headers and 31, 32
- secured connections for web 192, 193
- securing 164
- setting default reporting 23–24
- setting TCP/IP ports for ReportCast 24
- setting up connections to 26–28
- specifying custom configurations for 28
- split ReportCast configurations for 198, 200
- starting and stopping ReportCast for UNIX platforms 22
- starting and stopping ReportCast for Windows platforms 21
- ServerVersion variable 148
- service startup script 22, 24
- services 2, 20
- Services box 22
- session timeouts 168
- Set command 119, 144–145
- setting ReportCast template dependencies 49
- setting up ReportCast 4
- setting user preferences 11, 48
- Show Search URL option 15
- Size variable 157
- small icons 160
- __SmallIconURL variable 114

- SmallIconURL variable 160
- Smart Searches 16
- SmartSearch displays 73
- Solaris systems
 - installing ReportCast 23
- ?Sort directive 81
 - issuing 84–86
- ?sort directive 85
- sort by clause 123
- sort fields 85, 124
- sort order
 - changing 123
 - setting 85, 123
- sorting
 - folder items 85, 122, 125
 - on multiple fields 124
 - reports in file lists 122, 123
 - strings 29
- source files
 - displaying HTML 64
- spaces in string values 104
- special characters 32
- split ReportCast configuration 198–200, 206
- standard directory 50, 55
- Standard Requester page 47
- StartDateAndTime variable 146
- StartHitNumber variable 151
- starting
 - e.Analysis application 16
 - ReportCast 20–23
- startreportcast.sh 24, 31
- StartTime variable 156
- startwebagent.sh 22, 165
- ?Status directive 81
- status codes
 - unauthorized access 164, 169
- status directives 80, 108–110, 111
- status information
 - deleting 81, 156
 - deleting requests for 109
 - displaying 47, 109
 - generating 80, 81
 - generating lists for 130
 - getting 11, 108, 155, 158
 - requests for timing out 6
- status keyword 121
- Status template 62
- STATUS.ACHTML 47
- StatusText variable 158
- StatusURL variable 155, 156
- ?StdRequest directive 81, 98
- stopping ReportCast 20–23
- stopreportcast.sh 22
- storing non-Actuate objects 41
- strings
 - assigning to variables 145
 - character matching in 123
 - sorting 29
 - spaces in 104
 - special characters in 32, 33
- style sheets 83, 90
- subfolders 9
- ?Submit directive 81, 102–108
- Submit buttons 103
- submit parameter variables 104
- submit URLs 103
- ?submitCreation directive 35, 82, 112
- ?submitDetails directive 35, 82, 112
- ?submitForm directive 35, 82, 111
- ?submitSearch directive 83, 96
- submitting requests
 - for DHTML searches 83, 96
 - for report generation 10, 81, 98, 102, 103
- submitting subscription forms 35
- __submitUserPreferences directive 82
- Subscribed variable 160
- subscribing to ReportCast channels 4
 - getting subscribed users 135
 - process for 17
 - testing current subscriptions 160
 - with URLs 114
- subscribing to web channels 34
- subscription forms 35, 48, 82
- Subscription page 17
- subscription variables 115
- subtemplates
 - overview 49
 - renaming 57
- Success template 62
- SUCCESS.ACHTML 48
 - source code for 64

Sun Solaris. *See* Solaris systems

syntax

- channel directives 111, 112, 113, 114
- channel URLs 113
- Factory directives 98
- file directives 96, 97
- GetDynamicData directive 87
- GetReportData directive 89
- GetStyleSheet directive 90
- if command 142
- include command 143
- information directives 110, 111
- list command 119
- ReportCast commands 119
- ReportCast directives 78, 83
- requestSearch directive 95
- scripting variables 145
- searchReport directive 91, 93
- set command 144, 145
- sort and direction directives 85
- status directives 108, 109
- Submit directive 102, 103
- Versions directive 87
- ViewDefault directive 91
- ViewFrameset directive 86
- ViewNavigation directive 90
- ViewPage directive 88
- ViewTOC directive 90
- with command 143

syntax conventions (documentation) xvii

system dates/time 146

system-wide defaults 149

T

tab separated values 92

Table of Contents

- displaying for DHTML reports 90
- displaying reports from 88
- getting 83
- listing DHTML items in 138
- setting up 48
- testing for sub-items 150

taglines 104

TCP/IP port mapper

- connections to 193

- overview 190

- registering ports with 191

TCP/IP ports 24

TCP/IP protocol 189, 197, 203

template files 46

- caching 146

- creating 55

- defining variables in 145

- getting names 147

- listed 47

- loading 143

- naming 61

- overview 46

- reloading 60, 61

TemplateLoadDateAndTime variable 146

TemplateName variable 147

templates 3, 77

- accessibility 49–51

- adding to Encyclopedia 63, 64

- associating objects with 53

- changing 56, 60

- examples for 67, 69

- copying 29

- custom logins/logouts 167

- customizing 55, 58, 67

- examples 67–71

- elements described 55

- mapping directives to 62

- overview to ReportCast 46, 49

- referencing subtemplates in 57

- renaming 57

- search paths 50, 58

- searching for 57–59

- setting directory paths for custom 29

- setting directory structures 53

templates directory 49

temporary directories 42, 43

Text variable 150

third-party analysis tools viii

time

- getting current 146

- getting elapsed 146, 156

- getting for status information 158

- getting start 146, 156, 157

- last modification 152, 154

- setting default 154

time formats 105

- timeout intervals
 - connections 39
 - default 6
 - idle user session 168
 - requests 40
 - __timeToDelete variable 107
- titles 152
- TOC item objects 52
- TOC item variables 150
- Tocitems keyword 122
- Today variable 146
- TotalNumberOfHits variable 151
- TotalPages variable 149
- translating escape sequences 32
- TSV format 92
- tuning ReportCast performance 39–41
- two-way connections 189
- Type variable 158
- typographical conventions xvi
 - syntax xvii

U

- UDP protocol 195
- unauthorized access 43
- unauthorized status codes 164, 169
- unauthorized users 164, 188, 203
- uncompressed DHTML formats 88
- Uniform Resource Locators *See* URLs
- UNIX environments
 - accessing ReportCast 31, 32
 - configuring data reply ports 195
 - configuring network ports 189, 191
 - configuring viewing options for 42
 - enabling login authentication 167
 - formatting ReportCast directives for 78
 - implementing security for 164, 165, 166
 - installing ReportCast 23
 - mapping locales for 31
 - setting cached connections 192
 - setting connection timeouts for 39
 - setting default server 24
 - setting ports for 27, 28
 - setting request timeouts for 40
 - starting and stopping ReportCast on 22
 - unsubscribing from ReportCast channels 17
 - unsubscribing from web channels 34

- updates xi
- updating
 - Encyclopedia items 7
 - ReportCast channels 38
- URL variable 147, 153
- URLs
 - accessing Report Encyclopedia with 5
 - as ReportCast directives 76
 - assigning to images 37
 - building custom search 93
 - command separator for 85
 - custom configurations with 28
 - default reporting web site 3
 - displaying 77
 - displaying ReportCast channel 148
 - for detail pages 156
 - for Secure Sockets Layers 78
 - for submitting requests 103
 - getting admin 146
 - getting folder items 153
 - getting for canceled notifications 154, 156
 - getting for current channel 160
 - getting for current request 155, 156
 - getting for custom icons 160
 - getting for output 154, 156
 - getting for report executable 154
 - getting for report executables 156
 - getting for scheduled requests 148
 - getting for status page 155, 156
 - getting full 147
 - getting most recent file version 153
 - getting parent folder 152
 - getting previous 147
 - masking 33
 - setting view preferences from 11
 - specifying report location with 39
 - specifying server port in 26
 - translating escape sequences for 32
 - verifying search 15
- Use Web Server File System option 41
- user authentication
 - bypassing 166
 - default 164, 169, 170
 - enabling 181
 - failing 169
 - overriding 171, 175
 - process described 170, 173

- user interfaces 20
- user names
 - generating lists of 131, 134, 157
 - getting 147, 152, 155
 - setting 166
 - setting for notifications 106
- user preferences
 - changing 80
 - getting 149
 - setting 11, 48
- user preferences directives 80
- user preferences form 80, 82
- UserAgent variable 147
- user-defined report generation requests 127, 129
- UserName variable 147
- USERPREFERENCES.ACHTML 48
- __users variable 106
- users
 - accessing home folder for 81
 - changing passwords 182
 - displaying current 147
 - enabling idle session timeouts for 168
 - generating notification lists for 131, 134, 157
 - getting active requests 126
 - getting admin status for 148
 - getting available reports 34
 - getting granted permissions 132, 133
 - getting subscribed channels 35, 135
 - logging in as anonymous 166
 - logging in as different 167
 - logging in with different name 167
 - setting open connections for 39
 - simultaneous requests 195
 - testing for subscription status 160
 - unauthorized 164, 188, 203

V

- Value variable 151, 158, 159
- values keyword 121
- ValuesFileName variable 157
- ValuesObject constant 144
- variable substitution 145
- variables
 - assigning values 144

- directive-specific 104, 114, 115
- initializing 143
- scripting language 145–161
- <version> parameter 80
- version numbers
 - getting 146, 147, 148, 153, 156
 - in file names 32
 - specifying 33, 98
- __versionname variable 106
- ?Versions directive 81, 83, 87
- versions
 - getting available 81, 83, 124
 - getting specific report 98
 - testing 153
- versions keyword 121
- ?View directive 82, 97
- View Frameset page 48
- view process ports 190, 196, 197
- view session objects 52
- view session variables 149
- ?ViewDefault directive 82, 91
- Viewer x, 190, 198
 - downloading 12
- ViewFormatDefault variable 149
- ViewFormatPreference variable 149
- ?ViewFrameset directive 82, 86
- Viewframeset template 62
- VIEWFRAMESET.ACHTML 48
- viewing
 - available channels 112
 - channel contents 34, 48, 80
 - current users 147
 - DHTML reports 12, 48, 82, 86–91, 196
 - Encyclopedia contents 16, 80, 81, 125
 - factory requests 82
 - folder contents 8, 16, 47, 84, 125
 - folder lists 125
 - HTML source files 64
 - information for requests 47
 - navigation bar 90
 - online documentation xiii
 - PDF files 196
 - report files 86
 - reports 2, 16, 81, 96
 - reports in native file systems 41
 - reports in secured environments 193, 196
 - reports in specific formats 82

- viewing (*continued*)
 - reports with Actuate LRX 198
 - request details 110
 - status information 47, 109
 - Table of Contents 90
 - URLs 77
- viewing directives 81
- viewing options 11–12, 42, 48
- viewing performance 40–41
- viewing preferences 11, 48, 149
- Viewnav template 62
- VIEWNAV.ACHTML 48
- ?ViewNavigation directive 83, 90
- ?ViewPage directive 82, 88
- ViewTemp directory 42
- ?ViewTOC directive 83, 90
- Viewtoc template 62
- VIEWTOC.ACHTML 48
- ViewURL variable 154, 156
- viruses 188

W

- __wait variable 104, 106
- __WebAgent folder 49
- web browsers *x*, 77
 - accessing Encyclopedia with 4, 5
 - communications processes for 201, 203
 - communications protocol for 78
 - displaying link information 77
 - displaying links 77
 - getting host name for 147
 - getting IP addresses 147
 - issuing ReportCast directives from 76
 - loading reports 34
 - redirecting 26
 - refresh intervals 160
 - registering plug-ins for 32
 - reloading pages 7
 - resetting unauthorized status 169
 - secured connections for 192, 193
 - viewing reports with 2
- web casting 34
- web channels 34
- web environments
 - accessing e.Reporting Server 2, 169
 - accessing Report Encyclopedia 5–8

- deploying reports over 4, 188, 200
 - getting server information 147
 - implementing security 164
 - security configuration options for 201
 - setting up firewalls for 201
- web page templates 3, 46
- web pages
 - accessing Report Encyclopedia 5
 - changing border colors 70
 - checking dates of 6
 - creating 46
 - customizing 63, 64
 - customizing copyright information for 71
 - documentation and 4
 - generating 118, 145
 - getting available reports for 34
 - getting current 6
 - masking URLs for 33
 - predefined 47
 - reloading 7, 167
 - viewing document source for 64
- web request objects 51
- web request variables 147
- web server objects 51
- web server variables 147
- web servers 2
 - communications protocol for 78
 - configuring specific ports for 26, 27, 28
 - default authentication for 167
 - defining configuration-specific templates 50
 - native files systems for 41–43
 - resetting port numbers 26
 - response headers and 31, 32
 - secured connections for 192, 193
 - split ReportCast configurations for 198, 200
 - starting and stopping ReportCast for UNIX platforms 22
 - starting and stopping ReportCast for Windows platforms 21
- web sites
 - accessing files from non-default ports 26
 - accessing reports on Windows 32
 - Actuate home page *xi*
 - default reporting 2
 - securing 33

- webmaster 4
- <webserver> parameter 79
- WebServerVersion variable 147
- wildcards 123
- WINAPI calling convention 184
- Windows environments
 - accessing reports 32
 - changing ReportCast Server 25
 - changing TCP/IP ports for 24
 - configuring data reply ports 195
 - configuring network ports 189, 191
 - configuring viewing options for 42
 - enabling login authentication 167
 - implementing security for 164, 166, 184, 186
 - mapping locales for 29
 - setting cached connections 192
 - setting connection timeouts for 39
 - setting default server 23
 - setting ports for 26, 27
 - setting request timeouts for 40
 - setting viewing performance for 40, 41
 - starting and stopping ReportCast on 21
 - tuning viewing performance 40–41
- With command 119, 143–144
- World Wide Web *See* Web
- worms 188

X

- XML files
 - supporting locales for 31
 - view process ports for 190
 - viewing reports as 196
- XML style sheets 90
- XMLDisplay format 90

Z

- zooming percentage 87